

UNIVERSITE BLIDA-1

Faculté de Technologie

THESE DE DOCTORAT

en Electronique

**NŒUD DU RESEAU DE CAPTEUR SANS FIL (WSN) DANS UNE
ARCHITECTURE EMBARQUEE DE TYPE FPGA**

Par

Billel BENGHERBIA

Devant le jury composé de :

M. BENSEBTI	Professeur, U. de Blida	Président
M. DJEBARI	Professeur, U. de Blida	Examineur
S. SADOUDI	MCA EMP, Alger	Examineur
M. BOUHEDDA	MCA, U. de Médéa	Examineur
M. OULDZMIRLI	Professeur, U. de Médéa	Directeur de thèse
A. GUESSOUM	Professeur, U. de Blida	Co-Directeur de thèse

Blida, Juin 2018

الملخص

في هذا العمل، قمنا بتطوير عقدة استشعار لاسلكية قائمة على منصة FPGA مناسبة لنظام مراقبة الاهتزاز في الآلات وتشخيص الأعطال. وتستند عقدة الاستشعار اللاسلكية المقترحة على منصة FPGA من نوع Artix-7 XC7A35T Xilinx، متصلة عن طريق واجهتي اتصال تسلسلي SPI مع وحدتين: وحدة الإرسال والاستقبال اللاسلكية nRF24L01+ منخفضة التكلفة ومنخفضة الطاقة، بالإضافة إلى متسارع رقمي MEMS من نوع "ADXL 345". وفيما يتعلق بالأمن، قمنا بتصميم الأجهزة المعمارية لوحدات التشفير / فك التشفير من نوع AES-128 والتي يتم تنفيذها كنواة بملكية فكرية مخصصة لغرض تأمين البيانات المرسل. وبالإضافة إلى ذلك، قمنا بتحسين دقة التزامن عند الحصول على البيانات في وقت واحد في الشبكة. وقد اقترح حل على أساس عقدة التجميع التي تلعب دور المشرف لتحريك عمليات الاستحواد البيانات. من أجل التحقق من صحة العقد المتقدمة، تم إجراء الاختبارات التجريبية على منصة اختبار اختلال التوازن في آلة دوارة. وتظهر النتائج التي تم الحصول عليها أن العقد المصممة تستوفي متطلبات الحصول على البيانات المتزامنة حيث يبلغ متوسط الخطأ أقل من 83.33 نانو ثانية. وبفضل التشغيل الدوري لعقدة الاستشعار، تم تمديد العمر إلى أكثر من 48 ساعة من التشغيل. وقد تم تقدير هذه الوقت مع بطارية ذات قدرة نموذجية من 1000 ميلي امبير ساعي.

الكلمات المفتاحية: AES، FPGA، عقدة استشعار، شبكات الاستشعار اللاسلكية، أنظمة على الرقاقة، مراقبة، التزامن.

RESUME

Dans le présent travail, nous avons développé un nœud capteur sans fil à base d'un circuit FPGA pour un système de surveillance vibratoire des machines, dans le but de faire un diagnostic de défauts. Le nœud capteur sans fil proposé est basé sur le circuit FPGA « Artix-7 XC7A35T » de Xilinx, connecté via deux interfaces séries SPI avec deux modules : le module radio « nRF24L01+ » de faible coût et de faible puissance, et un accéléromètre numérique MEMS de type « ADXL 345 ». En ce qui concerne la sécurité, nous avons conçu deux architectures matérielles (niveau RTL) des modules de cryptage/décryptage AES-128 bits (Advanced Encryption Standard), qui sont implémentés comme des IP cores personnalisés, de sorte que les données transmises sont hautement sécurisées. En plus, nous avons amélioré la précision de la synchronisation lors d'une acquisition simultanée de données dans le réseau. Une solution a été proposée basée sur un nœud sink qui joue le rôle d'un maître pour déclencher les acquisitions de données. Pour valider les nœuds développés, des tests expérimentaux ont été effectués sur un banc d'essai d'une machine tournante ayant un défaut de déséquilibre. Les résultats obtenus ont montré que ces nœuds assurent les exigences d'acquisition de données synchrones où une erreur moyenne de 83.33 ns est atteinte. Grâce à un fonctionnement périodique du nœud capteur, la durée de vie a été élargie à une durée de plus de 48 heures d'exploitation possible. Cette durée a été estimée pour une batterie de capacité typique de 1000 mAh.

Mots clés : AES, FPGA, Nœud capteur, RCSF, SoC, Surveillance, Synchronisation.

ABSTRACT

In the present work, we have developed an FPGA based wireless sensor node for a machine vibration monitoring system and fault diagnosis. The proposed wireless sensor node is based on Xilinx's "Artix-7 XC7A35T" FPGA circuit, connected via two SPI-series interfaces with two modules: the RF transceiver nRF24L01+ as low-cost and low-power wireless radio module and a MEMS digital accelerometer type "ADXL 345". As far as security is concerned, we have designed two hardware architectures of the AES-128 (Advanced Encryption Standard) encryption/decryption modules, which are implemented as custom IP cores, so that the data transmitted is highly secure. In addition, we have improved the accuracy of synchronization when a simultaneously acquiring data in the network. A solution has been proposed based on a sink node which plays the role of a master to trigger data acquisitions. In order to validate the developed nodes, experimental tests are performed on a rotating machine test bench with unbalance fault. The results obtained shows that the designed nodes meet the requirements for synchronous data acquisition where an average error of less than 83.33 ns is reached. Thanks to a periodic operation of the sensor node, the lifetime has been extended to more than 48 hours of operation. This time has been estimated with a battery of typical capacity of 1000 mAh.

Key words: AES, FPGA, Sensor node, WSN, SoC, Monitoring, Synchronization.

REMERCIEMENTS

Je commence mes mots, par remercier ALLAH le tout puissant, pour m'avoir aidé et guidé pour atteindre ce point et terminer ce travail de doctorat.

Puis, je tiens à remercier mon directeur de thèse, le professeur Mohamed OULD ZMIRLI, pour m'avoir confié ce sujet de thèse, et sans lequel cette thèse n'aurait pas pu être concrétisée. Je le remercie pour ses conseils, sa gentillesse et sa disponibilité durant toutes ces longues années.

Mes sincères remerciements vont à mon co-directeur de thèse, le professeur Abderrezak GUESSOUM, pour son aide continue, sa disponibilité, et son support.

Je remercie profondément, Monsieur Messaoud BENSEBTI Professeur à l'université de Blida qui m'a fait l'honneur de présider ce jury.

Je suis reconnaissant aux membres du jury, le professeur Mustapha DJEBARI de l'université de Blida, le docteur Said SADOUDI de l'Ecole Militaire Polytechniques d'Alger, et le docteur Mounir BOUHEDDA de l'université de Médéa, pour avoir accepté de juger ce travail.

Je remercie, spécialement, le Dr. Mabrouk HAMDACHE pour m'avoir aidé à la rédaction de mon manuscrit.

A la mémoire de mon père Puisse Dieu, le tout puissant, l'avoir en sa sainte miséricorde

Un grand merci à ma mère pour leur soutien et leurs prières, ainsi qu'à tous les membres de ma famille.

Cette section sera incomplète sans avoir remercié tous mes amis et frères qui se reconnaîtront, dont le support moral, était ma source d'énergie dans l'accomplissement de ce travail.

SOMMAIRE

REMERCIEMENTS	4
SOMMAIRE	5
LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX	8
LISTE DES SYMBOLES ET ABREVIATIONS	10
INTRODUCTION GENERALE	12

CHAPITRE 1 : RESEAUX DE CAPTEURS SANS FIL

1.1 Introduction	18
1.2 Réseaux de capteurs sans fil	19
1.3 Les applications des RCSFs	25
1.4 Classification des RCSFs	29
1.4.1 Application avec un ensemble réduit de données et une faible complexité de calcul	30
1.4.2 Application avec un ensemble moyenne et élevé de données, et une complexité de calcul	30
1.4.3 Application avec des données et une complexité de calcul intensives	31
1.5 Les nœuds capteurs sans fil : Architecture et réalisation	31
1.5.1 Sous-système d'acquisition	33
1.5.2 Sous-système de traitement	33
1.5.3 Sous-système de communication	42
1.5.4 Sous-système d'alimentation	44
1.6 Topologies du réseau	45
1.7 Conclusion	48

CHAPITRE 2 : SYSTEMES SUR PUCE A BASE DE CIRCUITS FPGAs

2.1 Introduction	49
2.2 Description générale des systèmes embarqués	50

2.3 Technologie des processeurs embarqués	52
2.4 Circuits logiques programmables	56
2.4.1 SPLDs et CPLDs	57
2.4.2 FPGAs	59
2.5 Systèmes sur Puce	63
2.6 Co-design : Conception conjointe matérielle/logicielle	67
2.7 Flot de conception conjointe matérielle/logicielle	69
2.8 Flot de conception FPGA	72
2.9 Intérêt de l'approche SoC pour les nœuds de RCSF	74
2.10 Méthodologie de conception conjointe proposée pour les nœuds capteurs sans fil	74
2.11 Conclusion	77

CHAPITRE 3 :
CONCEPTION ET IMPLEMENTATION DES NŒUDS D'UN RCSF SUR UNE
PLATEFORME RECONFIGURABLE DE TYPE FPGA

3.1 Introduction	78
3.2 Travaux connexes	79
3.2.1 Nœuds capteurs avec un circuit FPGA autonome	80
3.2.2 Nœuds capteurs basés sur une combinaison d'un microcontrôleur et d'un circuit FPGA comme coprocesseur	82
3.2.3 Nœuds capteurs à base d'un circuit FPGA pour le prototypage des systèmes sur puce	84
3.3 Description des nœuds sans fil réalisés	85
3.3.1 Architecture globale du nœud	89
3.3.2 L'unité de contrôle	90
3.3.3 Bus et protocole AXI	92
3.3.4 L'unité de traitement FFT	92
3.3.5 L'unité de sécurité	95
3.3.6 Module de communication sans fil	103
3.3.7 Module d'acquisition des données	106
3.4 Résultats d'implémentation matérielle des nœuds	108
3.5 Conclusion	111

CHAPITRE 4 :
EVALUATION DES PERFORMANCES ET DE DIAGNOSTIC DES DEFAUTS
DANS UNE APPLICATION INDUSTRIELLE

4.1 Introduction	112
4.2 Evaluation des performances	112
4.2.1 Acquisition de données à des taux d'échantillonnage élevés	113
4.2.2 Temps de réponse des tâches réalisées par le nœud capteur	114
4.2.3 Consommation d'énergie du nœud capteur sans fil	116
4.2.4 Synchronisation temporelle des nœuds capteurs	118
4.3 Application pour la surveillance vibratoire des machines tournantes	122
4.3.1 Surveillance par analyse vibratoire	124
4.3.2 Problèmes fréquemment diagnostiqués dans les machines tournantes	124
4.3.3 Les défauts balourds	125
4.3.4 Description du banc d'essai	126
4.3.5 Tests pratiques de diagnostic de défaut balourd	128
4.4 Conclusion	132
CONCLUSION GENERALE ET PERSPECTIVES	134
BIBLIOGRAPHIE	138

LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX

Figure 1.1: Architecture générale d'un réseau de capteurs sans-fil	20
Figure 1.2: Exemples d'applications des RCSFs	29
Figure 1.3 : Architecture typique d'un nœud capteur sans fil	32
Figure 1.4 : Topologies du réseau ZigBee	45
Figure 2.1: Architecture typique d'un système embarqué	51
Figure 2.2: Croissance de la technologie des processeurs chez Intel	54
Figure 2.3 : Les différentes familles des circuits logiques programmables	57
Figure 2.4 : Une structure CPLD générique	59
Figure 2.5 : Structure interne de l'FPGA basée sur l'architecture de Xilinx	61
Figure 2.6 : Architecture interne d'un SoC typique	65
Figure 2.7 : Méthodologie de conception conjointe matérielle/logicielle	71
Figure 2.8 : Flux de conception des systèmes embarqués typiques	73
Figure 2.9 : Conception conjointe d'un nœud de capteur sans fil	76
Figure 3.1: La plateforme Digilent Basys-3	88
Figure 3.2: Architecture des nœuds sans fil : (a) nœud capteur; (b) nœud Sink	89
Figure 3.3: Diagramme en bloc du MicroBlaze	91
Figure 3.4: Le bloc radix-2 (a) radix-2 DIF butterfly (b) radix-2 DIT butterfly	93
Figure 3.5: Architecture "Pipelined Streaming I/O" du bloc FFT	95
Figure 3.6: L'organigramme de l'algorithme AES	97
Figure 3.7: Diagramme en blocs du module de cryptage réalisé	100
Figure 3.8: Diagramme en blocs du module de décryptage réalisé	101
Figure 3.9: Résultats de simulations AES: (a) Cryptage (b) Décryptage	102
Figure 3.10: Temps de traitement : (a) opération de cryptage (b) opération de décryptage	104
Figure 3.11: Organigramme de communication (a) Mode Tx (b) Mode Rx	106
Figure 3.12: Diagramme des blocs du nœud capteur	109
Figure 3.13: Diagramme des blocs du nœud Sink	110
Figure 4.1: Représentation spectrale pour différentes fréquences du signal d'entrée (a) 500 Hz (b) 1kHz (c) 5 kHz (d) 10 kHz	115
Figure 4.2: Temps de réponse des tâches réalisées par le nœud capteur	116

Figure 4.3: Durée de vie du nœud en fonction de la périodicité de mesure	117
Figure 4.4: Organigramme du procédé de synchronisation	120
Figure 4.5: Erreur mesurée commise dans le procédé de synchronisation	121
Figure 4.6: Résultats des tests d'erreurs de synchronisation mesurées (a) 1 ^{ère} expérience (visibilité directe) (b) 2 ^{ème} expérience (présence d'obstacles)	122
Figure 4.7: Les différentes méthodes d'analyse	124
Figure 4.8: Les méthodes d'extraction des caractéristiques basées sur les vibrations	125
Figure 4.9: Signal vibratoire généré par le balourd	126
Figure 4.10: Composition du banc d'essai utilisé	127
Figure 4.11: Emplacement de l'accéléromètre sur le palier	128
Figure 4.12: Système global de surveillance réalisé	129
Figure 4.13: Spectre de fréquence du signal mesuré par les nœuds capteurs et reçue par le sink. Fréquence de rotation : (a) 21,6 Hz; (b) 31,01 Hz	131
Figure 4.14: Spectre de fréquence du signal mesuré par les nœuds capteurs et reçue par le sink. Fréquence de rotation : (a) 40,5 Hz; (b) 45,2 Hz	132
Tableau 1.1: Evolution des nœuds capteurs	19
Tableau 1.2 : Résumé des capteurs utilisés dans les réseaux de capteurs sans fil	34
Tableau 1.3 : Exemples de nœuds des réseaux de capteurs sans fil	38
Tableau 1.4 : Attributs de diverses normes d'interface sans fil	46
Tableau 2.1: Comparaison des architectures	56
Tableau 2.2 : Comparaison des processeurs soft-core	67
Tableau 3.1: Nœuds capteurs avec un FPGA autonome	81
Tableau 3.2: Nœuds capteurs basés sur une combinaison d'un microcontrôleur et d'un circuit FPGA comme coprocesseur.	83
Tableau 3.3: Nœuds capteurs à base d'un circuit FPGA pour le prototypage des systèmes sur puce	86
Tableau 3.4: Ressources du circuit FPGA utilisé dans la plateforme BASYS-3	88
Tableau 3.5: Table Rcon utilisée dans l'extension des clés	98
Tableau 3.6: Consommation de ressources des IP-cores de cryptage/décryptage	103
Tableau 3.7: Comparaison des modules RF	105
Tableau 3.8: Résultats de synthèse et d'implémentation des nœuds	108
Tableau 4.1: Résultats de la consommation d'énergie du nœud pour les deux modes de fonctionnement	117
Tableau 4.2: Résultat de diagnostic pour deux fréquences d'échantillonnage.	130

LISTE DES SYMBOLES ET ABREVIATIONS

ADC	:	Analog to Digital Converter
AES	:	Advanced Encryption Standard
ASIC	:	Application Specific Integrated Circuit
ASIP	:	Application Specific Instruction Set Processors
AXI	:	Advanced eXtensible Interface
CAO	:	Conception Assistée par Ordinateur
CLB	:	Configurable Logic Block
CMOS	:	Complementary Metal Oxide Semiconductor
CPLD	:	Complex Programmable Logic Device
DSP	:	Digital Signal Processor
FFT	:	Fast Fourier Transform
FPGA	:	Field Programmable Gate Array
GPIO	:	General - Purpose Input/Output
GPP	:	General Purpose Processors
IC	:	Integrated Circuit
IP	:	intellectual property
ISM	:	Industrial, Scientific, & Medical radio frequency band
MAC	:	Media Access Control
MCU	:	Micro Controller Unit
MEMS	:	Micro-Electro-Mechanical System
PLD	:	Programmable Logic Device
QoS	:	Quality Of Service

RCSF	:	Réseau de Capteur Sans Fil
RF	:	Radio Frequency
ROM	:	Read Only Memory
RTL	:	Register-Transfer Level
RTOS	:	Real-Time Operating System
SoC	:	System On a Chip
SPI	:	Serial Peripheral Interface
SRAM	:	Static Random Access Memory
UART	:	Universal Asynchronous Receiver/Transmitter
USB	:	Universal Serial Bus
VHDL	:	VHSIC Hardware Description Language
VLSI	:	Very Large Scale Integration
WLAN	:	Wireless Local Area Network
WSN	:	Wireless Sensor Network

INTRODUCTION GENERALE

Le développement rapide que le monde de l'industrie a connu et l'évolution des processus industriels modernes se sont accompagnés d'une complexité dans les structures et les équipements mécaniques [1] qui, pour leurs parts, sont composés d'un grand nombre de pièces. Par conséquent, il est devenu important d'assurer la fiabilité des équipements industriels contre des pannes inattendues qui peuvent provoquer des pannes dans la chaîne de production industrielle, entraînant ainsi des conséquences économiques désastreuses pour les opérateurs économiques [2].

En raison du coût important de l'entretien périodique des équipements industriels, l'attention a été portée sur d'autres stratégies de maintenances préventives. Le choix est souvent orienté vers la maintenance conditionnelle qui est l'approche scientifique et pratique la plus efficace, notamment pour le cas des machines tournantes à travers la surveillance vibratoire de ces machines [3]. Ainsi grâce aux données collectées par une surveillance continue ou périodique, la maintenance conditionnelle est en mesure de fournir l'état réel de l'équipement. Elle permet de définir le temps idéal pour effectuer une réparation et améliorer la fiabilité et l'efficacité opérationnelle des machines et des équipements [4].

L'utilisation de la solution câblée dans les systèmes traditionnels de surveillance et de diagnostic des défauts trouve un succès dans plusieurs sites industriels [2][5]. Cependant, cette solution souffre de plusieurs inconvénients, d'abord en raison de son coût élevé d'installation et de maintenance, de la nécessité d'un câblage supplémentaire ou du remplacement des anciens câbles [6][7]. De plus, cette solution câblée n'est pas appropriée pour certains environnements industriels spécifiques, comme dans le cas d'environnements restreints, où la présence de

câbles peut provoquer des accidents humains et matériels outre la faible mobilité imposée par cette solution.

Avec l'émergence des réseaux de capteurs sans fil (RCSF, en anglais WSN) au début des années 1990 et le développement rapide des capteurs et des dispositifs MEMS, l'avenir des RCSFs devient de plus en plus prometteur et ouvre des voies à de nombreuses applications dans le monde réel, allant de la domotique jusqu'aux applications industrielles plus complexes [8]. Les chercheurs estiment un taux de croissance de 50% par an dans le marché des systèmes embarqués pour les applications des RCSFs [9] lié à ce type d'applications.

A l'heure actuelle, les RCSFs étendent de plus en plus leurs champs d'existence dans les applications de maintenance conditionnelle en raison des avantages attractifs qu'ils présentent, tels que la capacité de traitement intelligent, le faible coût, la simple installation, la flexibilité du réseau et la faible consommation d'énergie [10]. Une nouvelle classe des RCSFs conçue spécifiquement pour des applications industrielles a vu le jour pour répondre aux exigences de ce domaine, également appelé un RCSF Industriel ou IWSN [11].

Actuellement, la technologie des RCSFs a résolu avec succès les problèmes d'utilisation liés à la technologie câblée utilisée dans le contrôle de l'état des équipements. Cela a permis d'assurer une mobilité plus élevée et de réduire les coûts d'installation et de surveillance, ce qui s'est révélé comme une solution idéale pour un environnement industriel restreint et améliore le niveau de surveillance [12]. Quant à la surveillance vibratoire des machines, de nombreuses études basées sur les RCSFs ont été mises en œuvre dans des produits commerciaux, où ces solutions sont axées sur des problèmes d'efficacité énergétique, de la synchronisation et des algorithmes de traitement de données liés au type d'application comme la Transformée de Fourier Rapide (FFT) [13][14]. Les éléments constituant les RCSFs, appelés nœuds, sont, en général, basés sur des microcontrôleurs (MCU). Ces MCUs ont une capacité de calcul limitée associée à une faible consommation d'énergie, donc, ils ne sont pas en mesure d'exécuter des tâches de traitement de signaux complexes [15]. Pour remédier à ce problème,

plusieurs chercheurs ont proposé des solutions alternatives à base des MCUs spécialisés et plus puissants, comme les processeurs DSP et ARM [16].

Au cours des dernières années, nous avons assisté à plusieurs développements dans l'électronique intégrée, en particulier dans les systèmes de communication sans fil et l'apparition des circuits FPGA à faible consommation d'énergie et à faible coût. Ceci a permis aux nœuds et aux réseaux de capteurs sans fil d'être plus optimisés avec des performances plus élevées qui ont fait évoluer les RCSFs très rapidement. Dans les applications classiques, le rôle des circuits FPGA était limité aux applications de traitement du signal et à l'analyse des paquets réseau, alors qu'actuellement, le circuit FPGA se compose d'un certain nombre de ressources qui fonctionnent à grande vitesse telles que les slices DSP et les mémoires rapides, ce qui en fait un choix judicieux pour l'implémentation des nœuds capteurs de surveillance de vibrations des machines [17].

En plus de la limitation du calcul intensif, la synchronisation de l'acquisition est l'un des problèmes les plus connus dans ce type d'applications et qui a eu aussi sa part dans les travaux de recherche. Il a été résolu par Yuan et al. [5] en utilisant un récepteur GPS et par Araujo et al. [18] en combinant deux modules de transmission en même temps. Ces deux solutions sont limitées par le fait d'introduction d'une consommation d'énergie supplémentaire au système. Une autre solution pour remédier à ce problème de synchronisation de l'acquisition des données a été rapportée par Huang et al. [19], à l'aide d'une conception inter-couches « cross-layer design » des couches transversales. L'erreur de synchronisation moyenne obtenue par cette solution varie de 87 ns jusqu'à 190 ns dans le pire des cas.

Par ailleurs, un autre problème est à relever. Il a trait à l'aspect de la sécurité de transmission qui représente l'un des paramètres essentiels de l'application industrielle des RCSFs [20][21] dont le but est de rendre la communication sécurisée contre toutes attaques et intrusions externes. L'importance de la sécurité dans ces réseaux est devenue plus exigeante après son intégration dans les protocoles des IWSNs. Ainsi la norme wirelessHART assure la sécurité en utilisant un cryptage de type AES-128 bits à la couche MAC [22]. Dans la littérature, nous avons constaté que la sécurité des RCSFs industriels n'a pas été prise en considération dans la plupart des systèmes de surveillance proposés, en raison des ressources limitées des nœuds capteurs [20], et du calcul intensif requis par ce type

de protocole qui affecte considérablement, la plupart du temps, la qualité du service [23].

Falk et Hof [24] ont décrit les différentes architectures de sécurité spécifiquement conçues pour l'environnement industriel et ils ont déterminé les exigences de sécurité pertinentes en fonction d'une analyse de risques et de menaces. Islam et al. [25] ont analysé les défis de sécurité des RCSFs et leur fonctionnalité pour l'adoption industrielle. Ils rapportent qu'en assurant la fiabilité et en surmontant les faiblesses de la sécurité dans les communications sans fil, les réseaux renforceront leur acceptabilité en tant que technologie fiable pour l'utilisation dans les environnements industriels.

Pour tous ces motifs cités, la conception des nœuds capteurs pour les applications industrielles devrait surmonter plusieurs défis tels que le temps de traitement qui devrait être réduit sans influencer la qualité de service (QoS), étant donné que les systèmes de contrôle dans de telles applications sont généralement basés sur des algorithmes avancés qui nécessitent beaucoup de ressources matérielles et logicielles.

- Contributions

L'objectif principal de notre travail est le développement des nœuds sans fil d'un RCSF pour un système de surveillance vibratoire. Les différents nœuds, nœuds capteurs et nœud sink, sont implémentés sur des plates-formes FPGA basées sur un processeur soft-core MicroBlaze de Xilinx. L'adoption de la conception SoC peut être considérée comme une approche prometteuse pour améliorer les performances de ce type de nœuds, en particulier pour les applications industrielles rigoureuses. Elle offre une bonne solution pour le traitement des signaux acquis en temps réel.

Les principales contributions de cette thèse consistent en :

- Une étude du potentiel d'intégration de la logique programmable dans l'architecture des nœuds capteurs sans fil ainsi que l'utilisation du matériel reconfigurable considérée à plusieurs niveaux d'abstractions, allant du choix de la

technologie à l'utilisation dans les applications du monde réel. La thèse couvre également de multiples disciplines complémentaires telles que la conception de systèmes embarqués, le calcul reconfigurable, le traitement du signal et la cryptographie.

- Une implémentation en langage VHDL (niveau RTL) des modules de cryptage et de décryptage AES128 bits. Au niveau de la conception, nous avons exploité le concept du parallélisme dû à la grande dépendance inter-blocs. En plus, nous proposerons une architecture en pipeline adaptée aux besoins des nœuds capteurs en termes de calcul intensif et de temps de traitement. Cette contribution aura comme finalité le développement des IP-cores de cryptage/décryptage AES, suivant une conception hybride matérielle/logicielle. La partie logicielle est implémentée sur un processeur soft-core (MicroBlaze).

- Une amélioration de la rapidité de traitement, qui minimise l'erreur de synchronisation des nœuds capteurs.

- Organisation de la thèse

Ce manuscrit est élaboré en deux parties essentielles : la première se compose de deux chapitres sur l'état de l'art. La deuxième partie présente quant à elle, des solutions que nous avons proposées dans ce travail ainsi que les résultats de simulation, d'implémentation et des tests pratiques. Cette partie est également organisée en deux chapitres.

Partie 1 :

Chapitre 1 : il est consacré à une présentation générale des réseaux de capteurs sans fil, à leurs caractéristiques, ainsi qu'à leurs applications associées. Nous présenterons également les principaux composants d'un nœud capteur sans fil, son architecture et sa réalisation et ce dans le but de pouvoir proposer une nouvelle architecture des nœuds sans fil.

Chapitre 2 : ce chapitre présente une étude des systèmes sur puce à base de circuits FPGA. On entame le chapitre par une vue d'ensemble des systèmes embarqués et par la conception des systèmes numériques embarqués. Nous introduisons ainsi, la technologie des processeurs embarqués, l'évolution des

systèmes numériques et les circuits logiques programmables. Ensuite, nous présentons les systèmes sur puce (SoC) ainsi que le flot de conception conjointe matérielle/logicielle. Enfin, nous introduisons la méthodologie de conception proposée dans cette thèse.

Partie 2 :

Chapitre 3 : Il est dédié au développement matériel et logiciel des nœuds sur une plate-forme reconfigurable de type FPGA. En premier lieu, on expose les travaux connexes sur les nœuds capteurs sans fil à base des FPGAs dans le but de déterminer les différentes architectures possibles. Par la suite, nous décrivons en détail les différentes unités et modules qui constituent les nœuds réalisés. A la fin du chapitre, nous présentons les résultats de la partie implémentée sur une plateforme FPGA.

Chapitre 4 : Il présente l'évaluation des performances des nœuds proposés. Au début, nous décrivons la solution retenue pour assurer la synchronisation de l'acquisition ainsi que les résultats obtenus par des tests pratiques. Ensuite nous présentons les tests expérimentaux réalisés dans le cadre d'une application de surveillance vibratoire ainsi que les résultats des différents tests élaborés.

Ce document se termine par une conclusion générale où il est question des travaux réalisés, des résultats obtenus et d'un ensemble de perspectives.

CHAPITRE 1 :

RESEAUX DE CAPTEURS SANS FIL

1.1 Introduction

La technologie des réseaux de capteurs sans fil (RCSF) est une technologie clé pour l'avenir. Ces réseaux ont été identifiés comme l'une des technologies les plus importantes pour le 21ème siècle [26][27], et constituent encore un domaine de recherche relativement nouveau vu le développement continu de la technologie des semi-conducteurs, des techniques pour réduire la dissipation d'énergie et la technologie d'intégration dans les systèmes sur puce (SoC : System-on-a-Chip) qui permet de réduire le coût élevé de ces dispositifs, ainsi que leur taille par rapport aux premiers systèmes qui ont été proposés dans les réseaux de capteurs sans fil [28]. Le tableau 1.1 résume l'évolution des RCSFs. Une grande partie des recherches sur les RCSFs a traité de nouveaux domaines d'application visant à soutenir notre style de vie moderne. Des dispositifs intelligents et économiques dotés de plusieurs capteurs intégrés, mis en réseau via des réseaux sans fil et Internet et déployés en grand nombre, fournissent des opportunités sans précédent pour l'instrumentation et le contrôle des maisons, des villes et de l'environnement [26].

Dans ce chapitre, nous traiterons du cadre théorique sur lequel nous nous sommes basés pour traiter les différents aspects de conception des nœuds capteurs sans fil. Ce chapitre est scindé en deux parties : la première est consacrée à une présentation générale des RCSF, leurs différentes caractéristiques, les différents domaines d'applications ainsi qu'une proposition de classification des RCSF selon les objectifs visés par cette thèse. L'objectif de la deuxième partie de cette étude est de montrer quelques variantes de nœuds sans fil avec les différents modules qui le composent. Ensuite, une discussion sur les principales topologies des RCSF est présentée. Une conclusion clôturera ce chapitre.

Tableau 1.1: Evolution des nœuds capteurs

Caractéristiques	1^{ère} génération	2^{ème} génération	3^{ème} génération
Chronologie Taille	1980 – 1990 Boîte à chaussures ou plus grande	2000 – 2003 Paquet de cartes	2010 Particules de poussière
Poids	Kilogramme	Gramme	Négligeable
Architecture	Unité de Détection, traitement et communication séparés	Unité de Détection, traitement et communication intégrés	Unité de Détection, traitement et communication intégrés
Topologie	Point à point, étoile	Pair à pair, serveur client	Pair à pair
Source d'alimentation Durée de vie	Grandes batteries Heures à jours	Pile AA Jours à semaines	Energie solaire Des mois à des années
Déploiement	Placé par véhicule ou air drop	Mis à la main	Intégré

1.2 Réseaux de capteurs sans fil

Les RCSFs appartiennent à la famille générale de réseaux de capteurs ad hoc [27], c'est à dire un réseau sans infrastructure qui se compose de plusieurs minuscules nœuds sans fil (centaines, voire des milliers), également appelés motes ou capteurs distribués pour récupérer des données provenant de différents environnements d'intérêt. Ces motes sont extrêmement efficaces en matière de consommation d'énergie et peuvent communiquer efficacement avec d'autres motes. Ils peuvent être stationnaires mais aussi bien être placés sur une cible mobile, tout en étant dépendants des besoins de l'application. Une vue globale des réseaux de capteurs est illustrée à la figure 1.1.

Les nœuds capteurs sont répartis dans un champ de captage. Chacun des nœuds du réseau a la capacité de collecter des données et de les acheminer jusqu'à la station de base, ici le coordinateur du réseau sans fil ou puits (sink en anglais). Le nœud coordinateur peut être considéré comme un nœud racine dans une structure arborescente qui, dans certains réseaux, est conçu pour être un nœud de grande

capacité possédant des ressources matérielles et énergétiques plus importantes que le reste des nœuds capteurs du réseau. Son rôle est de stocker toutes les informations collectées par les nœuds capteurs. Toutes les communications des nœuds du réseau seront transmises au monde extérieur à travers le nœud coordinateur. Les données sont acheminées vers le coordinateur via l'architecture multi-saut du réseau sans fil. Le coordinateur peut communiquer avec le nœud operateur (utilisateur) directement ou à distance via un réseau intermédiaire (Internet ou satellite).

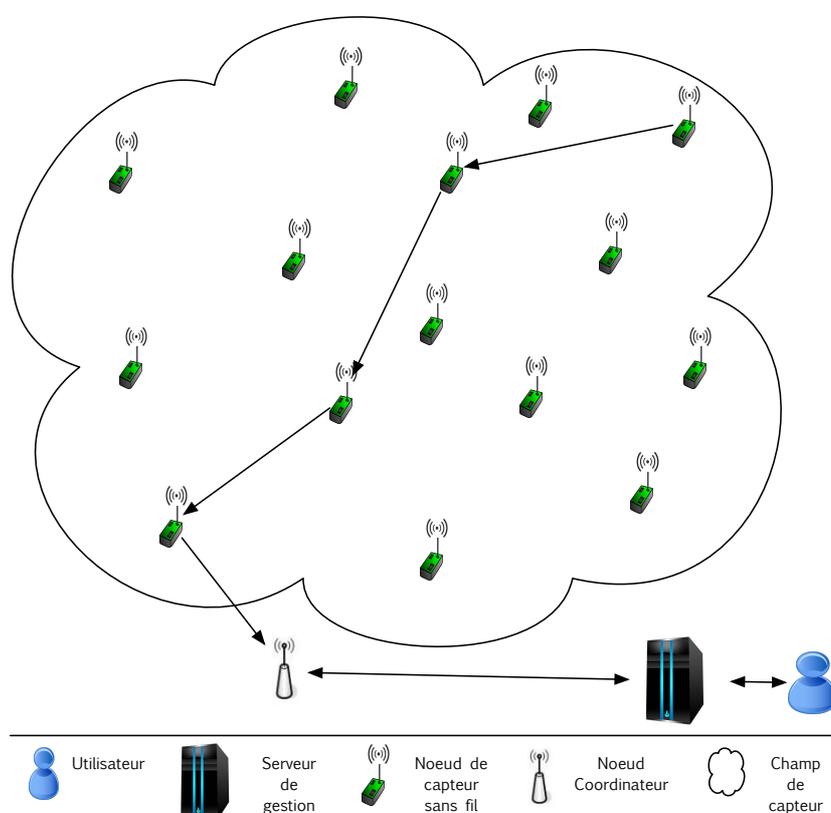


Figure 1.1: Architecture générale d'un réseau de capteurs sans-fil

Les RCSFs varient considérablement selon leurs caractéristiques [29]: la mobilité, ils peuvent être homogènes ou hétérogènes en termes de capacité de détection, ils peuvent détecter leur emplacement ou non, ils peuvent récolter de l'énergie ou dépendent uniquement des piles, ils peuvent être très simples ou possèdent une capacité de calcul complexe et ils peuvent être longtemps vécus ou jetables. Les principales caractéristiques d'un RCSF sont les suivantes :

- Déploiement de nœud [30]: Le réseau de capteurs peut être déployé au hasard dans une zone géographique. Après le déploiement, ils peuvent être maintenus automatiquement sans présence humaine. Dans le déploiement du nœud d'un réseau de capteurs, il y a deux catégories : soit un déploiement dense, soit un déploiement dispersé. Dans un déploiement dense, nous avons un nombre relativement élevé de nœuds capteurs dans le domaine ciblé tandis que dans un déploiement dispersé, nous avons moins de nœuds. Ce dernier est utilisé lorsque le coût des nœuds capteurs augmente et interdit l'utilisation d'un déploiement dense. Le déploiement dense est utilisé lorsqu'il est important de détecter chaque instant ou lorsque nous avons plusieurs capteurs pour couvrir une zone.
- Capacités de calcul [31]: C'est un autre facteur qui affecte les RCSFs en raison de la limite du coût, de la taille et de la consommation d'énergie de la batterie. L'espace du programme et l'espace mémoire du capteur sont très limités, ce qui rend la capacité de calcul et de traitement dans les nœuds capteur très limitée.
- Efficacité énergétique [32][33]: Le nœud capteur sans fil est un dispositif microélectronique, équipé d'un nombre limité de sources d'alimentation. L'énergie dans un RCSF est utilisée à des fins différentes telles que le calcul, la communication et le stockage. Le nœud capteur consomme plus d'énergie que d'autres pour la communication. S'ils manquent de puissance, ils deviennent souvent invalides et par conséquent, abandonnés, car nous n'avons aucun moyen de recharge. Pour cette raison, les chercheurs se concentrent sur la conception de protocoles et d'algorithmes qui devraient tenir compte de la consommation et de la conservation d'énergie dans la phase de conception et ce afin de maximiser la durée de vie du réseau de capteurs sans fil. Aussi, lors de la conception du réseau de capteurs, il faudrait tenir compte de son économie en énergie. Les composants tels que le microcontrôleur, le contrôleur de puissance et l'unité de communication devraient être conçus pour consommer moins d'énergie.

- Capacités de communication [34]: les différents nœuds dans un RCSF communiquent habituellement à l'aide d'ondes radio sur un canal sans fil où la bande passante de communication du réseau de capteurs est étroite et dynamique et sa distance de communication est seulement d'une dizaine à plusieurs centaines de mètres (courte portée). Le canal de communication peut être bidirectionnel ou unidirectionnel. Dans un environnement opérationnel autonome et hostile, la communication entre les nœuds sera facilement influencée par l'impact de l'environnement comme les montagnes, les bâtiments, les obstacles du terrain et la météo, ce qui rend difficile le maintien de la qualité de communication dans le réseau. En d'autres termes, le logiciel et le matériel de communication doivent être robustes et tolérants aux pannes, et assurer la sécurité, qui est un des secteurs de recherche importants pour l'avenir.
- Topologie évolutive, flexible et dynamique [34]: Dans le réseau de capteurs, le nombre de nœuds capteurs déployés peut être de l'ordre de centaines, de milliers ou de millions afin que nous puissions facilement étendre la taille du réseau. Sous certaines circonstances, des nœuds supplémentaires peuvent être ajoutés au réseau, ou même retirés des réseaux en raison de l'épuisement de la batterie et d'autres défaillances comme la perturbation du canal de communication. Cet état de fait entraînera des changements dans la topologie du réseau. Ainsi, contrairement aux réseaux traditionnels, où l'objectif est de maximiser le débit des canaux ou de minimiser le déploiement du nœud, dans le cadre d'un réseau de capteurs, il faut étendre la durée de vie du système et la robustesse du système. L'introduction de plus de nœuds dans le réseau signifie que des messages de communication supplémentaires seront échangés afin que ces nœuds soient intégrés au réseau existant, de sorte que la topologie du RCSF doit avoir la fonction de reconfiguration, de réglage dynamique et d'auto-ajustement. Les protocoles de communication doivent être conçus de telle sorte que le déploiement de plusieurs nœuds dans le réseau n'affecte pas le regroupement et le routage. En d'autres termes, le réseau doit conserver sa stabilité.

- Auto-organisation [30][33]: Comme les nœuds du réseau sont déployés de façon inconnue dans un environnement sans surveillance et hostile, les différents nœuds du réseau doivent pouvoir s'organiser. Le nœud capteur, qui peut s'adapter de façon collaborative, effectue et distribue un algorithme, peut rapidement et automatiquement former un réseau indépendant après la connexion des nœuds.
- Communication multi-saut [30][33]: Un nœud capteur ne peut communiquer qu'avec les nœuds voisins directs. Si un nœud doit communiquer avec la station de base ou bien un autre nœud en dehors de sa zone de couverture, cela consiste à prendre l'aide d'un nœud intermédiaire via un chemin de routage en traversant la voie multi-saut. Dans ce cas, les nœuds agissent non seulement comme collecteurs de données et expéditeurs, mais aussi en tant que routeurs d'informations.
- Support sans fil propice aux erreurs [33]: Étant donné que les réseaux de capteurs peuvent être déployés dans des situations différentes, les exigences de chaque application peuvent varier considérablement. Nous devrions considérer que le milieu sans fil peut être fortement affecté par des environnements bruyants. Un attaquant interfère sciemment et cause suffisamment de bruit pour affecter la communication.
- Pertinence de l'application [31]: Les RCSFs sont différents des réseaux conventionnels et dépendent fortement des applications du secteur militaire, environnemental et industriel. Les nœuds sont déployés de manière aléatoire et étendue en fonction du type d'utilisation. Son rôle principal consiste à acquérir les données locales via un protocole de routage qui ne peut pas être appliqué efficacement à d'autres types de réseaux. La pertinence de l'application est une des questions importantes pour le RCSF.

- Tolérance aux pannes et adaptabilité : La tolérance de panne signifie le maintien des fonctionnalités de réseau de capteurs sans interruption en raison d'une défaillance du nœud capteur car, dans ce type de réseau, chaque nœud a une puissance d'énergie limitée, de sorte que la défaillance d'un seul nœud n'affecte pas la tâche globale du réseau. Les protocoles adaptés peuvent établir de nouveaux liens en cas d'échec de nœud ou de congestion de lien. Le réseau peut s'adapter en changeant sa connectivité en cas de défaut. Dans ce cas, un algorithme de routage bien efficace est appliqué pour modifier la configuration globale du réseau.
- Petite taille [31]: les nœuds capteurs sont généralement de petite taille avec la gamme restreinte. En raison de leur taille, leur énergie est limitée, ce qui rend la capacité de communication faible ainsi que leur puissance de calcul.
- Temps réel [32]: le RCSF traite des événements du monde réel. Dans de nombreux cas, les données du capteur doivent être livrées dans des limites de temps afin que des observations appropriées puissent être faites ou des mesures prises. Il existe très peu de résultats à ce jour concernant le respect des exigences en temps réel dans les RCSFs. La plupart des protocoles ignorent le temps réel ou tentent simplement de traiter les données le plus rapidement possible et espèrent que cette vitesse est suffisante pour respecter les délais.
- Qualité de service [25][35]: Cela signifie que les données devraient être livrées dans les délais. Certaines applications en temps réel sont basées sur le temps, ce qui signifie que les données doivent être livrées à temps à partir du moment où elles sont détectées, sinon les données deviendront inutilisables. Par exemple, la détection des incendies nécessite une bonne qualité de services.
- Sécurité [32][34]: La sécurité est un paramètre très important dans le réseau de capteurs puisque ces réseaux sont basés sur les données acquises, de

sorte qu'il n'y a pas d'identifiant particulier associé aux nœuds capteurs. L'attaquant peut facilement s'insérer dans le réseau, en interceptant des données importantes et en y devenant une partie du réseau sans la connaissance des nœuds capteurs du réseau. Il est donc difficile de vérifier si les informations sont authentifiées ou non.

1.3 Les applications des RCSFs

La popularité des nœuds capteurs sans fil à faible puissance est en expansion continue au cours de la dernière décennie, déclenchant ainsi une ère d'or pour la recherche et le développement de réseaux de capteurs sans fil [36]. Cette expansion est due principalement aux recherches qui ont été lancées dans de nombreux domaines scientifiques (comme la physique, la microélectronique, le contrôle, la science des matériaux, ... etc.) et la collaboration ciblée de scientifiques qui, traditionnellement, visaient des directions totalement différentes, et qui ont conduit à la création de systèmes micro-électromécaniques, communément appelé MEMS (Micro-Electro-Mechanical Systems) [37][30]. Les MEMS ont réussi à briser les limites de ce qui était considéré comme un système sur puce (SoC), qui étaient auparavant supposés ne comporter que des fonctions logiques, alors qu'actuellement la mesure des paramètres physiques et de l'actionnement devient possible grâce à l'intégration des capteurs et des actionneurs au silicium. En parallèle, avec le développement de l'industrie du silicium qui a fait des progrès étonnants, la technologie RF et les circuits numériques ont également progressé de façon spectaculaire. Les émetteurs-récepteurs à faible puissance et à fréquence plus élevée sont implémentés sur des puces, tandis que les circuits numériques ont tendance à être fabriqués de plus en plus denses. Tous ces progrès ont permis aux RCSFs de supporter aujourd'hui une large gamme d'applications, allant de la détection environnementale au suivi des véhicules, de la sécurité périmétrique à la gestion des stocks, et de la surveillance de l'habitat à la gestion des champs de bataille. Des exemples d'applications potentielles des RCSFs seront exposés par la suite (Figure 1.2):

- Applications militaires [38]: Considérées comme l'un des premiers domaines d'applications des RCSFs, qui sont étroitement liées à la notion de réseaux de capteurs sans fil, surtout avec leurs caractéristiques telles que le déploiement rapide, la tolérance aux pannes ainsi que l'auto-configuration qui font de ce type de réseaux un outil majeur dans un tel domaine. En fait, il est très difficile d'affirmer si les nœuds capteurs ont été développés en raison des besoins militaires ou s'ils ont été inventés de manière indépendante et ont ensuite été appliqués aux services de l'armée. En ce qui concerne les applications militaires, la zone d'intérêt s'étend, en général, de la collecte d'informations au suivi de l'ennemi, à la surveillance des champs de bataille, à la classification des cibles, à la détection des dangers comme les radiations ainsi qu'aux missions d'espionnages. Les applications en période de paix telles que la sécurité intérieure, la protection et la surveillance de la propriété, la patrouille frontalière, ...etc., sont des activités qui ont lieu aussi dans les réseaux de capteurs.
- Surveillance de l'environnement [39]: Une autre catégorie importante d'applications possibles a trait à ce qu'on appelle "suivi environnemental". Les RCSFs peuvent être utilisés pour surveiller et déterminer les valeurs de certains paramètres à un endroit donné, comme par exemple : la température, l'humidité et la pression atmosphérique, ...etc. En outre, un gaspillage majeur d'énergie se produit par un chauffage ou un refroidissement inutile des bâtiments. Un nœud capteur peut aider à utiliser des appareils de chauffage, des ventilateurs et d'autres équipements pertinents de manière raisonnable et économique, ce qui entraînera un environnement plus sain et un meilleur confort pour les résidents. Les nœuds capteurs peuvent être utiles dans la nature pour détecter des événements tels que les feux de forêts, les tempêtes ou les inondations, ce qui permet une intervention efficace et beaucoup plus rapide des services de secours.
- Applications médicales [33]: La science de la santé et le système de soins de santé peuvent également bénéficier de l'utilisation de capteurs sans fil. Les applications médicales des RCSFs visent à améliorer les services de santé et

de surveillance existants, en particulier pour les personnes âgées, les enfants et les malades chroniques. De nombreux avantages sont obtenus avec ces systèmes.

- Capacité de surveillance à distance. C'est le principal avantage des systèmes de soins de santé omniprésents. Avec la surveillance à distance, l'identification des conditions d'urgence pour les patients à risque devient facile et les personnes ayant des degrés différents d'insuffisances cognitives et physiques seront aidées à avoir une vie plus indépendante et plus facile. Les petits enfants et les bébés seront également pris en charge de manière plus sûre pendant que leurs parents sont absents.
- L'identification et la prise en compte en temps réel des systèmes de soins de santé généralisés sont parmi les principaux avantages. L'identification en quelques secondes ou minutes de situations d'urgence, comme les crises cardiaques, suffira pour sauver des vies, compte tenu du fait que, sans systèmes en temps réel, ces conditions ne seront pas identifiées. Les progrès technologiques dans l'électronique grand public ont réduit les coûts de production et ont permis de disposer de dispositifs capteurs peu coûteux pour les utilisateurs ordinaires.
- Être capable d'identifier le contexte est un autre avantage obtenu grâce aux systèmes de soins de santé omniprésents. La sensibilisation au contexte permet de surveiller constamment les conditions de la personne et les environnements dans lesquels elles se trouvent. Cette information contextuelle est obtenue principalement en détectant des systèmes qui intègrent plus d'un type de capacités de détection. En fusionnant les informations recueillies par plusieurs capteurs, une compréhension plus claire du contexte peut être obtenue. L'information contextuelle aide à mieux identifier les modèles inhabituels et à faire des inférences plus précises sur la situation. Par exemple, pendant la nuit, être dans la chambre à coucher en position couchée peut ne pas indiquer quelque chose de grave, alors que

le coucher au milieu de la journée peut indiquer une situation d'alarme. La connaissance du contexte fournit cette information utile.

- Applications industrielles [36]: L'application des RCSFs à la conception de réseaux de terrain pour les systèmes industriels de communication et de contrôle offre des avantages majeurs en termes d'installation et de maintenance souple d'appareils de terrain, de support pour le suivi des opérations de robots mobiles et la réduction des coûts et les problèmes liés au câblage. Aussi, les applications d'automatisation industrielle fournissent le contrôle, la conservation, l'efficacité et la sécurité, comme suit :
 - Étendent les systèmes de fabrication et de contrôle de fabrication existants de manière fiable.
 - Améliorent la gestion des actifs en surveillant continuellement les équipements critiques.
 - Réduisent les coûts d'énergie grâce à des processus de fabrication optimisés.
 - Permettent d'identifier un fonctionnement inefficace ou un équipement peu performant.
 - Automatiser l'acquisition de données à partir de capteurs distants pour réduire l'intervention de l'utilisateur.
 - Fournissent des données détaillées pour améliorer les programmes de maintenance préventive.
 - Aident à déployer des réseaux de surveillance pour améliorer la sécurité des employés et du public.
 - Aident à rationaliser la collecte de données pour améliorer les rapports de conformité.

Cette brève description de quelques domaines d'applications est loin d'être exhaustive car le nombre de domaines où les RCSFs sont utilisés est très vaste.

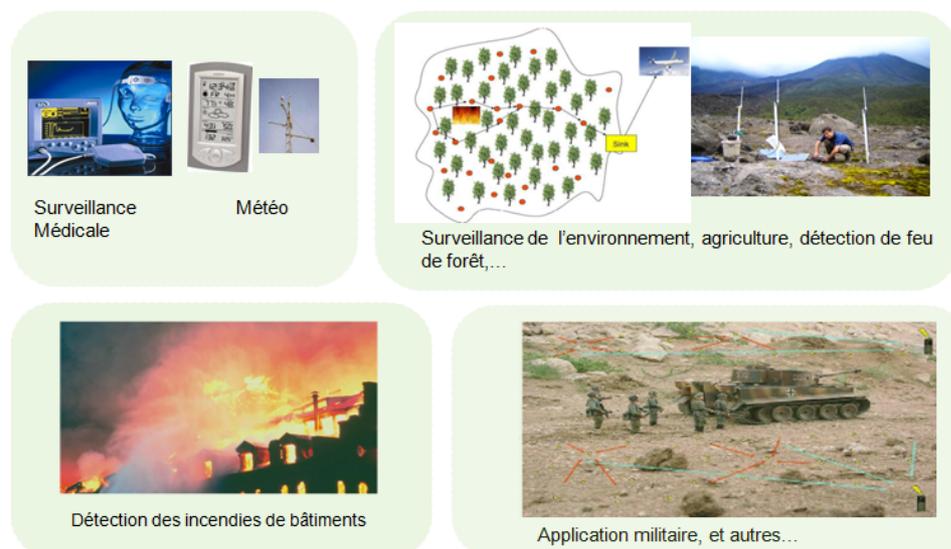


Figure 1.2: Exemples d'applications des RCSFs [40]

1.4 Classification des RCSFs

Afin d'évaluer les exigences et les défis associés à la conception et au déploiement des RCSFs pour une diversité d'applications comme mentionné ci-dessus, il est important de classer d'abord ces applications en fonction d'une certaine pertinence afin qu'un espace de conception puisse être déduit pour un ensemble donné d'applications. Plusieurs taxonomies ont été proposées dans la littérature pour classer les applications des RCSFs. Tilak et al. [41] ont défini une taxonomie anticipée qui permet la classification des RCSFs selon trois approches : les fonctions de communication, l'introduction des données dans le système de transmission ainsi que la dynamique de réseau. Une classification des RCSFs selon l'objectifs de l'application, les caractéristiques de trafic et les besoins en matière de livraison de données est présenté [42]. Karl et al. [43] présentent une classification, dans laquelle un nœud capteur coopère avec un collecteur de données. Ils ont classé les applications des RCSFs dans la surveillance, le suivi et la détection d'événements. Aucune des taxonomies proposées ne semble saisir totalement tous les aspects pertinents à la conception des RCSFs, et ne fournit pas les moyens de déduire un espace de conception optimisé pour l'application traitée dans cette thèse. Le travail mené par Shahzad [44] propose une classification des applications potentielles des RCSFs selon l'ensemble de données détectées et leur complexité des calculs afin d'explorer une architecture économe en énergie pour

les applications à forte intensité de données et de calcul. Leur taxonomie est également conçue comme un outil pour identifier les exigences d'application communes que les plates-formes des RCSFs doivent respecter. Les trois classes sont :

1.4.1 Application avec un ensemble réduit de données et une faible complexité de calcul

Un grand nombre d'applications de surveillance nécessitent des nœuds statiques, dont chacun ne surveille que peu de paramètres à une fréquence relativement faible. De plus, les données acquises, allant de quelques octets à des dizaines d'octets, sont souvent transmises sans une nécessité de traitement complexe outre qu'une simple agrégation (addition / soustraction). Des exemples de telles applications comprennent le suivi général de l'environnement, le suivi des cultures et des sols dans l'agriculture, le contrôle des gaz à effet de serre, le suivi de différents procédés industriels, ...etc.

1.4.2 Application avec un ensemble moyenne et élevé de données, et une complexité de calcul

Certaines applications des RCSFs, exigent des nœuds capteurs mobiles afin de relayer les données acquises. C'est le cas par exemple des applications de suivi et de la surveillance des véhicules et des troupes. Dans une telle application, la taille et la complexité de traitement des données peuvent souvent être faibles. Cependant, la fréquence d'acquisition et de transmission pourrait être irrégulière, ce qui modifie à un moment donné, la taille des données à traiter. Outre l'acquisition des données souhaitées à un taux d'échantillonnage faible, le soutien géographique et la nécessité de localiser les différents nœuds du réseau, pourraient augmenter la taille et la complexité des données à traiter. Les facteurs qui déterminent la taille des données et la complexité du calcul incluent les types et le nombre de capteurs utilisés, la fréquence à laquelle les données de chacun des capteurs sont acquises et le niveau de l'information qui doit être extraite des données brutes. Selon les objectifs et les exigences d'un scénario d'application donné, la quantité de données et la complexité impliquées dans le traitement de ces données peuvent être

résolues facilement avec un nœud capteur sans fil typique et à faible coût. Dans d'autres cas, ils peuvent nécessiter des ressources de traitement à haute performance et/ou des modules de communication pour réaliser de telles applications.

1.4.3 Application avec des données et une complexité de calcul intensives

Les meilleurs candidats aux applications à forte intensité de données sont les applications de surveillance qui acquièrent des données scalaires à un taux d'échantillonnage élevé, ou lorsque les données associées à chaque échantillon sont volumineuses, par rapport au débit de données limité des émetteurs-récepteurs qui sont généralement utilisés dans les nœuds capteurs sans fil. Une autre catégorie d'application à forte intensité de calcul nécessite un traitement complexe et intensif afin d'extraire des informations utiles à partir de la grande quantité de données acquises.

Un exemple de ces applications fait l'objet d'une étude dans cette thèse, c'est le cas des applications de surveillance industrielle. Dans une telle application, la méthode la plus répandue pour analyser l'état de fonctionnement de la machine est basée sur l'analyse des vibrations. En ce qui concerne les RCSFs, en particulier, la surveillance multiaxiale et à haute fréquence génère de grandes quantités de données et nécessite un traitement intensif du signal pour analyser ces données. De nombreuses études basées sur les RCSFs ont été mises en œuvre et des études sont axées sur des algorithmes efficaces de traitement de l'information dans ce type d'application, comme la Transformée de Fourier Rapide (FFT).

1.5 Les nœuds capteurs sans fil : Architecture et réalisation

Les nœuds capteurs sans fil communément appelés « motes » ou tout simplement nœuds sans fil, sont l'élément central d'un réseau de capteurs sans fil. C'est à travers un nœud que se déroule la majorité des fonctionnalités du réseau comme la détection, le traitement et la communication. Il stocke et exécute les protocoles de communication et les algorithmes de traitement de données. La qualité, la taille et la fréquence des données détectées qui peuvent être extraites du réseau sont

influencées par les ressources physiques disponibles pour le nœud. Par conséquent, la conception et la mise en œuvre d'un nœud sans fil est une étape critique. La recherche sur les nœuds sans fil est généralement focalisée sur la conception de différents composants du nœud pour permettre au système en général de fonctionner pendant la plus longue période possible dans des conditions extrêmes. Cela s'applique à différentes parties du système, y compris le matériel, le système d'exploitation, et les logiciels d'applications. La principale préoccupation dans la conception de la plupart des nœuds comprend la consommation d'énergie et la taille physique. La puissance de traitement et la mémoire, ainsi que les communications sans erreur sont d'autres contraintes auxquelles les nœuds doivent faire face et ils ont un impact supplémentaire sur les deux premiers facteurs.

L'architecture générale d'un nœud sans fil typique est illustrée à la figure 1.3. Les principaux composants du nœud consistent en des sous-systèmes de détection et d'acquisition, de traitement et stockage, de communication, et d'alimentation. Des fonctionnalités facultatives supplémentaires peuvent être incorporées dans la conception du nœud, comme le système de localisation.

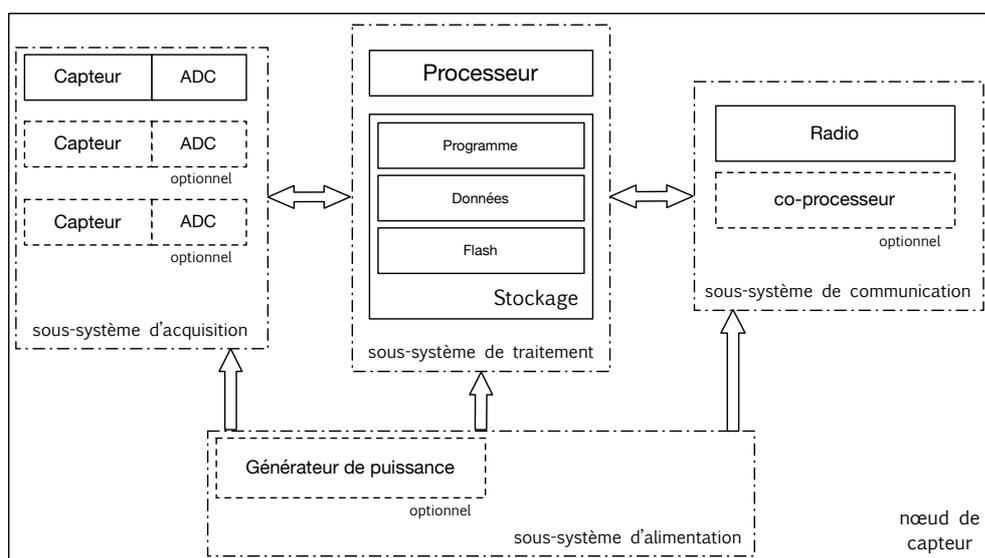


Figure 1.3 : Architecture typique d'un nœud capteur sans fil

Le choix des technologies pour l'implémentation des nœuds est généralement déterminé par les contraintes de conception imposées par l'application. Le concepteur a une multitude d'options pour décider comment concevoir et implémenter ces sous-systèmes en un nœud programmable et unique. Les sous-systèmes peuvent être physiquement indépendants, comme c'est le cas lorsque le nœud est conçu avec des composants disponibles sur le marché ou qu'ils peuvent être intégrés comme un SoC.

1.5.1 Sous-système d'acquisition

Un nœud peut inclure un ou plusieurs capteurs physiques selon l'application. Le capteur dans le sous-système d'acquisition, représente une interface entre l'environnement et le nœud sans fil. La détection des phénomènes physiques n'est pas quelque chose de nouveau, mais l'arrivée des systèmes MEMS a permis de mesurer et de quantifier les grandeurs physiques à faible coût. Un capteur physique contient un transducteur, qui est défini comme un dispositif qui transforme les grandeurs physiques observées en une grandeur électrique [45]. La sortie de ce transducteur est un signal analogique ayant généralement une grandeur continue en fonction du temps. Par conséquent, un convertisseur analogique-numérique est nécessaire pour interfacier un sous-système d'acquisition et de détection avec un processeur numérique. Le tableau 1.2 fournit un résumé détaillé des types de capteurs qui ont été utilisés dans les RCSFs. Il fournit également un résumé concis des grandeurs qu'ils capturent et des aspects de ces grandeurs. Cette liste est loin d'être exhaustive, mais elle met en évidence la portée et l'utilité des RCSFs et la richesse des capteurs qui peuvent être utilisés.

1.5.2 Sous-système de traitement [46]

Le sous-système de traitement est le contrôleur principal du nœud capteur sans fil. Il regroupe tous les autres sous-systèmes ainsi que certains périphériques supplémentaires. Son but principal est de traiter et exécuter les instructions relatives à la détection, à la communication et à l'auto-organisation.

Tableau 1.2 : Résumé des capteurs utilisés dans les réseaux de capteurs sans fil

Capteur	Domaine d'application	Grandeur détectée	Observation
Accéléromètre	Surveillance de l'état des structures [47][48]	Accélération 2D et 3D des mouvements de personnes et d'objets	Détection d'amplitude et évaluation de la fréquence naturelle. Evaluation de la fréquence naturelle et validation de la méthode de détection des dégâts.
	Soins de santé [49]		Mesurer les mouvements du corps en combinant à la fois des accéléromètres et des puces de gyroscope dans un seul nœud capteur sans fil.
	Surveillance industrielle [50]		L'unité de détection composée d'un accéléromètre effectue les mesures des vibrations des ailes sur la structure de l'avion.
Gyroscope	Wireless body sensor network [51]	Vitesse angulaire	Le système a été développé pour l'application d'analyse de la marche pour ceux qui souffrent des maladies du Parkinson et d'Alzheimer en utilisant une unité de mesures inertielles intégrées dans un réseau de capteurs sans fil.
Capteur de fréquence cardiaque	Soins de santé [52]	La mesure du rythme cardiaque	Le battement de cœur est détecté, un signal radio est transmis, dans lequel le récepteur l'utilise pour déterminer la fréquence cardiaque actuelle. Le signal sera une impulsion radio simple ou un signal codé unique de la sangle de poitrine.

Capteur magnétique	Systèmes de transport intelligents [53][54]	Présence, intensité, direction et variation du champ magnétique	Présence, vitesse et densité des véhicules dans une rue.
Capteur acoustique	Localisation et mobilité [55][56]	Vibration de la pression acoustique	Localisation des nœuds capteurs dans le réseau de capteurs acoustique sous-marin, dont les positions sont inconnues en raison de leur distribution aléatoire.
Capteur chimique	Surveillance de l'environnement [57]	détecter et surveiller les divers types de gaz	étudier l'impact de la dynamique des capteurs sur la précision de mesure et de la localité grâce à des expériences systématiques dans l'environnement contrôlé d'un tunnel de vent.
Capteur pH	Pipeline [58]	Concentration en ions hydrogène	Indique le contenu acide et alcalin de l'eau.

Le type d'unité de traitement dans un nœud est choisi en fonction de la classe d'application prévue ainsi que de la complexité de ses tâches, car cela affecte le coût, la flexibilité, les performances et la consommation d'énergie du nœud. Du point de vue matériel, le sous-système de traitement peut être de type microcontrôleur, de type FPGA ou les deux à la fois. D'autres options sont possibles, à savoir utiliser les ASICs et les DSPs.

Si la tâche de détection est bien définie dès le début et ne change pas avec le temps, un concepteur peut choisir soit un FPGA, soit un processeur de traitement de signal (DSP). Ces processeurs sont très efficaces en termes de consommation d'énergie ; pour la plupart des tâches de détection, ils sont tout à fait adéquats. Cependant, comme ce ne sont pas des processeurs à usage général, le processus de conception et l'implémentation peuvent être complexes et coûteux.

Les RCSFs sont des technologies émergentes. Le milieu de recherche est toujours actif dans le but de développer des protocoles de communication et des algorithmes de traitement de signaux performants en termes de rendement énergétique. Comme cela nécessite une installation et une mise à jour de code dynamique, le microcontrôleur est l'une des meilleures options.

Conduite par les progrès technologiques, une grande variété de nœuds a été développée comme solutions industrielles au cours des dernières années. Certaines de ces architectures sont également disponibles comme prototypes expérimentaux développées dans le cadre des études sur la conception des nœuds et des réseaux. Des exemples sur quelques nœuds sans fil ainsi que leurs éléments de traitement et de communication sont répertoriés sur le tableau 1.3.

1.5.2.1 A base des microcontrôleurs

Les premières versions des nœuds capteurs étaient basées sur les microcontrôleurs (MCU). Ces nœuds étaient principalement composés d'un émetteur-récepteur radio, d'une antenne, d'un processeur, d'une unité de mémoire, d'un capteur et d'une batterie. Un microcontrôleur peut être choisi en raison de la flexibilité de programmation qu'il offre. Sa construction compacte, sa petite taille, sa

faible consommation d'énergie et son faible coût. MSP430 et Atmel ATMega 128L sont deux microcontrôleurs populaires utilisés pour les nœuds capteurs sans fil. Ces MCU utilisent généralement le TinyOS comme système d'exploitation en raison de son efficacité énergétique.

Malgré les variances des nœuds basés sur le microcontrôleur et leurs fabricants, ils ont la même architecture de base. La différence réside dans les capacités des nœuds capteurs en termes de puissance du microcontrôleur, de capacité de mémoire et de communication radio.

L'utilisation de langages de programmation de niveau supérieur augmente la vitesse de programmation et facilite le débogage. Il existe des environnements de développement qui offrent une abstraction de toutes les fonctionnalités du microcontrôleur. Cela permet aux développeurs d'applications de programmer des microcontrôleurs sans avoir besoin d'une connaissance de bas niveau du matériel.

Cependant, les microcontrôleurs ne sont pas aussi puissants et efficaces, ce qui se résulte en un impact sur la vitesse de traitement et de transmission des nœuds. Par conséquent, la réflexion sur de nouvelles architectures qui satisfont aux limites des MCU a été une obsession importante pour les recherches, comme c'est le cas de certains processeurs personnalisés tels que les DSP et les FPGA.

1.5.2.2 A base desDSPs

Les algorithmes tels que le filtrage numérique, l'analyse de Fourier et les techniques similaires nécessitent des opérations spécifiques ainsi que des calculs numériques intensifs.

Ceci est particulièrement intéressant pour les applications des RCSFs de multimédia dans lesquelles le traitement du signal audio et vidéo dans le réseau peut être nécessaire pour compresser ou agréger des données de grande taille. Les processeurs généraux ne sont pas toujours optimisés pour ces types d'algorithmes.

Tableau 1.3 : Exemples de nœuds des réseaux de capteurs sans fil

Nœud (Référence)	Type de la plateforme	Détails de la plateforme	Émetteur-récepteur
Mica	Microcontrôleur	ATMega103 (8 bits, 4MHz)	TR1000
Mica2		ATMega128L (8 bits, 7.37MHz)	TI CC1000
MicaZ		ATMega128L (8 bits, 7.37MHz)	TI CC2420
Telos		MSP430 (16 bits, 8MHz)	TI CC2420
NFC-WISP [59]		MSP430 (16 bits, 8MHz)	RFID (13.56)
iMote2		Intel PXA271 (13 – 416 MHz)	TI CC2420
[60]		ARM7TDMI (32 bits, 50 MHz)	TI CC2420
[61]	DSP	TI TMS320C6412 (32bits, 500 – 750MHz)	TI CC2420
[62]		TI TMS320C6416T (32bits, 600MHz – 1GHz)	//
WINS[63]	ASIC	Analyseur de spectre	915MHz conception personnalisée
[64]		Microcontrôleur personnalisé	433MHz RF module
[65]		DSP personnalisé	RF module personnalisé
[66]	FPGA	Nios (softcore, 32bits, 60 MHz)	nRF2401
RecoNode[67]		Virtex-4 (PowerPC hardcore)	TI CC2520
[68]		Actel Igloo AGL1000 (20 MHz)	TI CC2420
HaLOEWEn[69]		Actel Igloo AGL100 (8052)	TI CC2531
RUSH [70]		Zynq 7020	Bluetooth
SENTIOF-CAM [71]	FPGA / Microcontrôleur	Spartan-6 – Atmel(32 bits, 60 MHz)	TI CC2520
PowWow[72]		Actel Igloo / MSP430	TI CC2420
[73]	FPGA / DSP	//	TI CC2480

En raison de leur flexibilité et de leurs performances dans le traitement du signal, les DSP sont utilisés comme une solution dans plusieurs architectures des RCSFs pour effectuer efficacement les calculs mathématiques du traitement de signal grâce à leur architecture matérielle et logicielle. Les DSPs sont également utiles pour les applications nécessitant le déploiement des nœuds dans des environnements physiques difficiles où la transmission du signal risque d'être endommagée en raison du bruit et des interférences.

1.5.2.3 A base des circuits ASICs

Un ASIC (Application Specific Integrated Circuit) est un circuit électronique qui peut être personnalisé pour une application spécifique et intègre sur une seule puce tous les éléments nécessaires pour réaliser des tâches spéciales. Contrairement à un microcontrôleur, un ASIC peut être facilement conçu et optimisé pour répondre à une demande spécifique. Aussi, C'est une solution rentable, car les interconnexions ainsi que la structure logique peuvent être spécifiées en fonction des besoins de l'utilisateur. Cela offre également une grande flexibilité et une réutilisabilité.

Le rôle le plus approprié des ASICs dans les RCSFs n'est pas de remplacer les microcontrôleurs ou les DSPs mais de les compléter. Certains sous-systèmes peuvent intégrer des processeurs personnalisés pour gérer des tâches élémentaires et de bas niveau et pour séparer ces tâches du sous-système principal de traitement. C'est le cas par exemple, de certains modules de communication qui sont livrés avec un noyau de processeur intégré pour améliorer la qualité des signaux reçus, annuler le bruit et effectuer des contrôles cycliques de redondance. Ces types de processeurs à usage spécial peuvent être efficacement réalisés en utilisant les ASIC.

Cependant, un ASIC présente certains inconvénients typiques comme les difficultés de conception, le manque de reconfigurabilité, le délai de mise sur le marché et les coûts de développement habituellement élevés. Aujourd'hui, les nouvelles FPGAs, à faible coût et à faible puissance, deviennent une alternative attractive pour

remplacer les ASICs, offrant ainsi des performances, des coûts et des délais de mise sur le marché.

1.5.2.4 A base des circuits FPGAs

La distinction entre un ASIC et un FPGA n'est pas toujours claire. En fait, l'anatomie de base des deux architectures est essentiellement la même. Les FPGAs sont plus complexes dans la conception et plus souples à programmer. Avec les vastes améliorations technologiques, les dispositifs programmables ont connu une grande évolution pour atteindre les performances qu'ils ont aujourd'hui. Les FPGAs et les CPLDs sont deux types connus de dispositifs programmables (plus de détails sur différentes plateformes seront présentés dans le deuxième chapitre).

Les FPGAs ont une bande passante plus élevée par rapport aux DSPs ; Ils sont plus flexibles dans leur application et peuvent supporter un traitement parallèle. Alors que les DSPs et les microcontrôleurs peuvent intégrer un ADC interne, un FPGA ne le fait pas. Semblable à un DSP, un FPGA a la capacité de travailler avec la représentation en virgule flottante. En outre, un FPGA expose sa vitesse de traitement aux développeurs d'applications, ce qui leur donne une plus grande souplesse de contrôle. Par rapport aux ASICs, l'utilisation d'un FPGA comme plateforme présente plusieurs avantages. Tout d'abord, les implémentations peuvent être mises à jour afin d'introduire une nouvelle fonctionnalité, même partiellement et en cours d'exécution. Deuxièmement, les FPGAs actuels pris en charge dans ces architectures de multiplieurs, des additionneurs et même des DSPs slices qui fonctionnent à haute fréquence. En outre, les FPGAs suivent la loi de Moore, c'est-à-dire que ces plates-formes basées sur un SRAM offrent les avantages des derniers processus de fabrication des circuits intégrés en termes de réduction de surface et de consommation d'énergie. Enfin, leurs coûts sont beaucoup plus faibles par rapport à la fabrication de circuits intégrés personnalisés.

Grâce à leur performance et à leur capacité de reprogrammation, les FPGAs attirent l'attention des développeurs des RCSFs. Ils sont employés dans des nœuds capteurs sans fil de 3 façons : des plates-formes autonomes, des plates-formes

modulaires et des sous-systèmes dans un système sur puce (SoC). De nombreux travaux sur les RCSF ont montré que l'FPGA a gagné l'acceptation rapide et a été un candidat de choix pour répondre aux exigences des RCSFs grâce à leur flexibilité et leur reconfiguration dynamique.

1.5.2.5 Discussion

L'utilisation des microcontrôleurs est préférable si l'objectif de conception est de chercher plus de flexibilité, et ne conviennent que pour le traitement des tâches simples. Par conséquent, ils ne peuvent pas satisfaire les exigences croissantes des RCSFs, notamment avec la forte croissance des technologies de pointe et la variété des applications. Alors que les microcontrôleurs ont une mémoire limitée, des progrès considérables sont réalisés afin d'augmenter leur taille. Récemment, de plus en plus de microcontrôleurs sont disponibles sur le marché avec des fonctionnalités intéressantes. L'utilisation avec les autres plateformes est préférable si la consommation d'énergie et l'efficacité de calcul sont souhaitées.

En comparaison, les DSPs sont coûteux, de grande taille et moins flexibles. En outre, les DSPs sont meilleurs pour le traitement du signal, avec des algorithmes spécifiques. Les ASICs ont une bande passante plus élevée ; Ils sont les plus petits en taille, exécutent beaucoup mieux et consomment moins d'énergie que n'importe quels autres types de plateforme. Le principal inconvénient est le coût élevé de la production vue la complexité du processus de conception. Les performances peuvent être améliorées grâce à l'application des systèmes multi-cœurs où plusieurs applications peuvent être exécutées en parallèle. Cela permet d'intégrer les ASICs dans les autres sous-systèmes, de sorte que, lorsque le sous-système du processeur principal est inactif et qu'il doit donc être désactivé, les tâches élémentaires peuvent être effectuées par des ASICs plus performants.

Les FPGAs sont plus rapides que les microcontrôleurs et les processeurs de signaux numériques et prennent en charge le calcul parallèle. Dans les réseaux de capteurs sans fil, l'FPGA a attiré beaucoup d'attention et a été un candidat privilégié pour satisfaire les exigences des RCSFs grâce à sa flexibilité. Cependant, la

difficulté de programmation et la forte consommation énergétique des FPGAs les rendent moins souhaitables. Ces inconvénients peuvent être surmontés par l'utilisation de flash FPGA. L'exemple est celui des FPGA Igloo d'Actel qui sont optimisés dans la consommation d'énergie et qui peuvent même fonctionner sur ultra-faible puissance.

1.5.3 Sous-système de communication

Le dispositif de communication est un composant important du nœud capteur sans fil qui permet l'échange de données entre les différents nœuds du réseau via un support de communication sans fil. Cette unité représente la partie matérielle la plus consommatrice d'énergie. Les différents choix de média de transmission incluent les ondes radio, la communication optique et les ultra-sons. D'autres médias comme l'inductance magnétique ne sont utilisés que dans des cas très spécifiques. De ces choix, la communication basée sur la radiofréquence (RF) est de loin la plus pertinente, car elle correspond le mieux aux exigences de la plupart des applications des RCSFs. Elle offre une portée relativement longue, des débits de données élevés, et des taux d'erreur acceptables avec un niveau de consommation énergétique raisonnable. Le type radiofréquence comprend des circuits de modulation/démodulation, filtrage, multiplexage/démultiplexage et autres dispositifs électroniques. Voici quelques exemples des normes les plus utilisées :

- Le Bluetooth : également connu sous la norme IEEE 802.15.1. Il est basé sur un système radio sans fil courte portée, peu coûteuse, avec une faible consommation d'énergie, un débit raisonnable et, par conséquent, adapté aux divers petits appareils à piles. Avec ces caractéristiques, les modules Bluetooth semblent répondre bien aux exigences des réseaux de capteurs sans fil. Le Bluetooth offre aux nœuds sans fil un débit jusqu'à 3 Mbps. Il a une portée d'environ 100 mètres et offre une durée de vie de quelques jours, plus longue que celle du WiFi. Il utilise une technique de saut de fréquence avec un duplex temporel (TDD) pour minimiser l'impact des interférences dans la bande ISM (2,4 GHz). Par ailleurs, il existe plusieurs problèmes, qui doivent être résolus avant que le Bluetooth puisse être déployé dans de

grands réseaux de capteurs, tels que le délai d'établissement des connexions et la taille limitée (8 nœuds, y compris le coordinateur) du réseau qu'il peut former. Des travaux sont en cours, et de nouveaux standards dérivés de Bluetooth sont développés, comme c'est le cas avec le BLE (Bluetooth LowEnergy) qui permet un débit du même ordre de grandeur pour une consommation dix fois moindre que le Bluetooth.

- Le Wi-Fi : (abréviation de Wireless Fidelity « Wi-Fi »). Il comprend les normes IEEE 802.11 a/b/g pour les réseaux locaux sans fil (WLAN). Le Wi-Fi utilise la radiofréquence pour permettre le transfert de données à grande vitesse sur de courtes distances. Il permet aux réseaux locaux de fonctionner sans câbles, ce qui en fait un choix populaire pour les réseaux domestiques et commerciaux. Il fournit également l'accès à internet haut débit sans fil pour la plupart des appareils modernes, tels que les ordinateurs portables, les téléphones mobiles, les PDA et les consoles de jeux électroniques. Pour les applications des RCSFs, le Wifi offre des caractéristiques attractives comme le débit élevé (56 Mbps) sur une bande de fréquence 2.4, une large bande passante et une longue portée (plus de 100 mètres). Toutefois, il est handicapé par sa grande consommation énergétique, ce qui limite la durée de vie des nœuds sans fil à quelques heures seulement empêchant ainsi son utilisation surtout pour les projets des RCSFs ayant de fortes contraintes en énergie sauf en version WiFi ultra-low power.
- Zigbee (IEEE 802.15.4) : C'est une norme définie pour les dispositifs de communication de données fonctionnant dans des réseaux locaux sans fil à faible débit (LR-WPAN). C'est la solution la plus utilisée dans les RCSFs puisqu'il fournit des communications à faible coût, à courte portée, à faible puissance et à faible débit. (250 kbps au maximum). Il cible les applications des capteurs sans fil, qui nécessitent une communication à courte portée pour maximiser la durée de vie d'un nœud capteur sur plusieurs années. De plus, il permet une taille de réseau plus dense (jusqu'à 65536 nœuds). Il fournit également des réseaux maillés auto-organisés, multi-saut et fiables. Le protocole IEEE 802.15.4 comprend une couche physique (PHY) et une

sous-couche MAC pour les réseaux LR-WPAN. La couche PHY offre trois bandes de fréquences opérationnelles ; Il existe 27 canaux alloués dans la gamme 802.15.4, avec 16 canaux dans la bande de 2,4 GHz, 10 canaux dans la bande de 915 MHz et 1 canal sur 868 MHz.

Il existe beaucoup de différences techniques et de similitudes entre Bluetooth, ZigBee et Wi-Fi. La principale différence inclut le débit de données, la consommation d'énergie, la taille du réseau, le coût, la couverture à distance et la sécurité. Le choix de la technologie à utiliser doit être basé sur l'application cible car chaque application des RCSFs a des exigences différentes sur le système de communication. Bien que certaines applications nécessitent une latence très faible, d'autres nécessitent une connexion haut débit et sécurisée ou une longue durée de vie de la batterie. Le développement de nouvelles technologies pousse les RCSFs dans de nouveaux domaines d'application. Alors que les technologies comme le WirelessHART et ISA100, rendent les RCSFs plus robustes dans le domaine de surveillance industrielle. Des technologies comme le BLE, ZigBee green power, Wi-Fi direct et EnOcean permettent aux RCSFs d'être utilisés dans d'autres secteurs en plein essor tels que celui des soins de santé, des automobiles, de la domotique, des sports et de l'agriculture. Une étude comparative de quelques technologies radio existantes pour les RCSFs est fournie dans le tableau 1.4.

1.5.4 Sous-système d'alimentation

Pour les nœuds capteurs sans fil, l'alimentation est un composant crucial du système puisqu'elle influe directement sur la durée de vie des capteurs et donc du réseau. Le sous-système d'alimentation peut avoir deux aspects essentiels :

- Stocker l'énergie et la fournir sous la forme requise. Dans ce cas, le stockage est effectué généralement dans des piles. Comme un ordre de grandeur, une pile AA à donne environ 2.2-2.5 Ah fonctionnant à 1.5 V.
- Dans les environnements sensibles où une plus longue vie de réseau est essentielle, il est impossible de recharger ou changer une batterie. Une

solution pour remédier à ce problème, est la reconstitution de l'énergie consommée par un réapprovisionnement à partir d'une source d'alimentation externe comme les cellules solaires, la vibration, ...etc.

1.6 Topologies du réseau [74][75]

Les réseaux Zigbee peuvent se connecter ensemble dans plusieurs configurations ou topologies différentes pour décrire la manière dont les périphériques du réseau sont connectés et ce afin de donner au réseau sa structure. Leur arrangement physique, bien sûr, peut être différent. Il existe trois topologies importantes de ZigBee, illustrées à la figure 1.4:

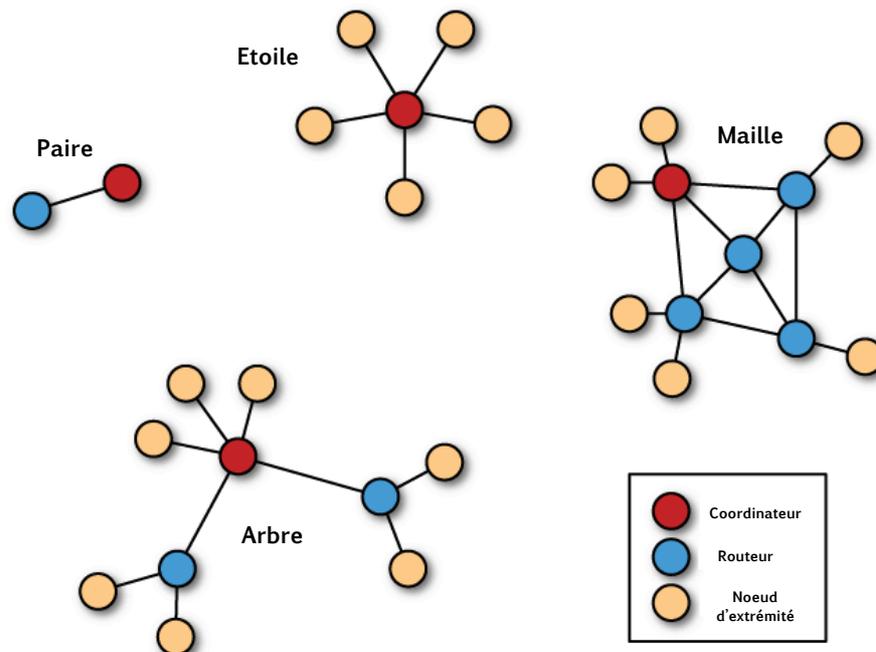


Figure 1.4 : Topologies du réseau ZigBee

- **Paire :** Le réseau le plus simple est celui avec seulement deux nœuds. Un nœud doit être un coordinateur afin que le réseau puisse être formé. L'autre nœud peut être configuré comme un routeur ou un nœud d'extrémité (End-Device). En général, les projets qui n'auront plus besoin d'une seule paire de nœuds n'obtiendront aucun avantage de ZigBee.

Tableau 1.4 : Attributs de diverses normes d'interface sans fil

	ZigBee [76]	Bluetooth [77]	BLE [34]	ISA100 [75]	WirelessHART [78]	EnOcean [34]	Wi-Fi [79]
Fréquence (ISM)	868 / 915 MHz, 2.4 GHz	2.4 GHz	2.4 GHz	2.4 GHz	2.4 GHz	868 / 315 MHz	2.4 GHz, 5 GHz
Débit Max.	250 kbps	3 Mbps	1 Mbps	250 kbps	250 kbps	125 kbps	54 Mbps
Couverture	100 m	10 – 100 m	200 m	75 m	200 m	300 m	100 m
Vie de la batterie	Jours – Années		Mois - Années			Sans batterie	
Topologie	Etoile, P2P ^a , Maille	P2P	P2P	Maille, Etoile, Cluster	Maille		
Consommation d'énergie	Faible	Faible	Très faible	Très faible	Très faible	Très faible	Très élevé
Nombre des nœuds Max.	>65000	8	8	250 (65000) ^b	250 (65000)	2 ³²	>10000

^a : Peer to peer

^b : 250 nœuds, c'est la limite pratique

- Topologie étoile : Une topologie en étoile comprend essentiellement un coordinateur central avec des connexions point à point à tous les autres périphériques qui peuvent être des routeurs ou des nœuds d'extrémités. Le coordinateur contrôle l'accès au réseau ainsi que tout échange de données entre les périphériques. Si le coordinateur échoue, le réseau échoue, alors que l'échec des lignes individuelles n'est pas critique. L'étendue du réseau est limitée par le nombre de connexions au coordinateur. En raison de l'énorme consommation d'énergie du coordinateur, cette topologie peut ne pas être adéquate pour les RCSFs, mais elle reste toujours préférable lorsque la zone de couverture est faible et une faible latence est requise par les applications des RCSFs.
- Topologie maille : C'est une topologie décentralisée qui se caractérise par sa capacité à fournir des chemins redondants à un dispositif central (station de base ou coordinateur). Chaque nœud du réseau peut recevoir des informations d'un nœud voisin et le transmettre à un autre. Ces nœuds sont appelés des nœuds de routage. Le maillage est auto-organisé, les chemins disponibles étant stockés dans des tables de routage. Lorsque certaines routes sont en panne et que le nœud de routage ne reçoit aucun accusé de réception d'un message renvoyé, il utilise un itinéraire alternatif parmi les routes disponibles. Afin d'assurer une transmission fiable, chaque nœud de routage doit être en contact avec au moins deux voisins. Plus les voisins d'un nœud de routage sont grand, plus leur qualité de signal est meilleure et plus l'intégrité du réseau est élevée. Le nombre de nœuds est théoriquement illimité, mais des restrictions peuvent être imposées par le protocole utilisé ainsi que la densité du trafic et les temps de réponse requis par l'application. Cette topologie consomme moins d'énergie par rapport à la topologie étoile. Le principal inconvénient de cette topologie réside dans la complexité de la mise en œuvre du routage.
- Topologie arbre : c'est un cas particulier qui combine les topologies étoile et maille. Un dispositif d'extrémité se joint à l'arbre par l'intermédiaire d'un

routeur. Ce dernier se joint à l'arbre par l'intermédiaire d'un autre routeur (le coordinateur ZigBee peut être utilisé aussi en tant que routeur). Chaque routeur maintient la synchronisation des échanges de données des dispositifs. Le reste du temps, les end-divices peuvent économiser l'énergie en mode veille.

1.7 Conclusion

Dans ce chapitre, nous nous sommes appliqués à la présentation des réseaux de capteurs sans fil. Cette catégorie de systèmes possède les caractéristiques susceptibles de les rendre comme une solution attrayante dans notre vie quotidienne. Les RCSFs ont constitué un axe de recherche très fertile dans plusieurs applications, comme la surveillance de l'environnement et dans le milieu industriel, le domaine militaire, la santé, la météorologie, la sécurité, ...etc. Cependant, il y reste encore de nombreux défis à relever afin de pouvoir les utiliser dans des cas réels. C'est le cas par exemple des applications de surveillance industrielle où la problématique majeure dans un tel système est le traitement intense des données acquises durant la surveillance, un problème difficile à surmonter par les nœuds capteurs ordinaires en raison de leurs limites en termes de capacité de calcul et de traitement.

Pour faire face à ce problème la mise en œuvre des RCSFs nécessite tout d'abord une étude préliminaire des technologies existantes afin de construire un système qui satisfait aux besoins génériques des nœuds sans fil ordinaires, ainsi que les besoins spécifiques à l'application de surveillance industrielle comme la capacité de calcul, la synchronisation des données et la sécurité du réseau.

Après avoir présenté les différentes technologies constituant les nœuds capteurs sans fil, le reste de cette thèse va s'intéresser spécialement à la présentation des systèmes sur puce à base du circuit FPGA. L'intérêt porté à celle-ci est dû à ses multiples avantages et spécificités qui ont rendu prometteuse son application dans le domaine des RCSFs.

CHAPITRE 2 :

SYSTEMES SUR PUCE A BASE DE CIRCUITS FPGAs

2.1 Introduction

Les progrès récents dans la technologie des circuits FPGAs, et l'amélioration continue de ses caractéristiques telles que la rapidité et la flexibilité, la grande quantité de ressources intégrées et la disponibilité de noyaux spécialisés de propriété intellectuelle (IP), le court délai de mise sur le marché, le bon compromis coût-performance, font des FPGAs la plate-forme d'implémentation préférée dans de nombreux domaines d'applications et ouvre la porte à de nouvelles fonctionnalités qui peuvent contribuer de manière significative au développement des systèmes numériques plus efficaces. Au fur et à mesure que les FPGAs ont évolué de manière significative en profitant de la réduction du coût de la technologie de fabrication, les fournisseurs ont commencé à développer des processeurs logiciels (soft-cores) qui peuvent être implémentés à partir de ressources FPGA standard, ainsi qu'à intégrer des processeurs matériels (hard-cores) dans leurs dispositifs. Cette tendance a connu un développement continu, dans la mesure où les solutions actuelles sont nombreuses.

De ce fait, l'ancienne approche de conception a abouti à un changement du paradigme qui constitue le principal atout actuel des FPGAs, qui ne peuvent plus être considérés comme de simples accélérateurs matériels, mais comme des plates-formes System-on-Chip (SoC) très puissantes.

Ce chapitre, se veut une entrée en matière dans le domaine de la conception des systèmes embarqués à base d'FPGA. Après une présentation générale sur les systèmes embarqués, une description générale des processeurs embarqués est donnée. Ensuite, les circuits logiques programmables et plus précisément les circuits FPGAs ainsi que les SoCs sont présentés. Enfin, une méthodologie de

conception conjointe d'un nœud de capteur sans fil est proposée. Elle fournit un flux de conception complet allant des spécifications jusqu'à l'implémentation finale sur le circuit FPGA.

2.2 Description générale des systèmes embarqués

Les systèmes informatiques sont partout, il n'est donc pas surprenant que des millions soient fabriqués chaque année et qu'ils sont destinés à être utilisés dans les ordinateurs personnels, les postes de travail, les ordinateurs centraux et les serveurs [80].

Les systèmes informatiques sont aussi de plus en plus intégrés dans différents appareils électroniques pour effectuer des fonctions particulières et qui sont souvent non reconnues par les utilisateurs de ces appareils.

Les systèmes embarqués ont été définis de plusieurs façons. Ci-après est rapportée une série de définitions de différents auteurs dans le domaine :

L'auteur dans [81] a défini le système embarqué comme un système qui intègre des logiciels embarqués et du matériel informatique, ce qui en fait un système dédié à une application ou à une partie spécifique d'une application ou d'un produit ou d'une partie d'un système plus large. Dans [80], l'auteur a défini un système informatique embarqué comme tout dispositif qui inclut un ordinateur programmable mais qui n'est pas lui-même destiné à être un ordinateur à usage général.

L'auteur de [82] quant à lui a défini les systèmes embarqués comme des systèmes électroniques qui contiennent un microprocesseur ou un microcontrôleur, mais on ne les considère pas comme des ordinateurs. L'ordinateur est caché ou intégré dans le système.

D'après l'auteur de [83], un système embarqué peut être défini comme un système basé sur un microprocesseur qui est conçu pour contrôler une fonction ou une plage de fonctions et n'est pas conçu pour être programmé par l'utilisateur de la même

manière qu'un ordinateur.

Ces définitions ne sont pas précises, mais elles peuvent être aussi proches que possible. Nous pouvons mieux comprendre ces systèmes en examinant des exemples. Une telle description montre les principaux défis auxquels sont confrontés les concepteurs de ces systèmes.

Un système embarqué est un système qui comporte trois composants principaux [81]:

- Il intègre du matériel similaire à un ordinateur. La figure 2.1 montre les principales unités qui composent le matériel d'un système embarqué. Comme son software s'intègre habituellement dans la ROM ou la mémoire flash, il n'a généralement pas besoin d'un disque dur secondaire comme c'est le cas dans l'ordinateur.

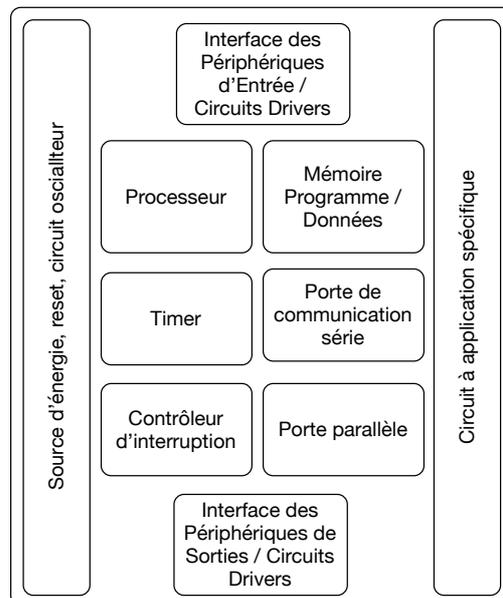


Figure 2.1: Architecture typique d'un système embarqué

- Il intègre le logiciel d'application principal dans le but d'exécuter simultanément une série de tâches ou de processus.
- Il intègre un système d'exploitation en temps réel (RTOS) qui supervise le logiciel d'application en cours d'exécution et organise l'accès aux ressources en

fonction des priorités des tâches du système. Il fournit un mécanisme permettant au processeur d'exécuter un processus tel que prévu et détermine le moment d'effectuer un changement de contexte entre les différents processus. Il définit les règles lors de l'exécution du logiciel d'application. Un système embarqué à petite échelle peut ne pas intégrer un RTOS.

Un système embarqué se caractérise par ce qui suit [80] [81]:

- Les opérations en temps réel et le traitement d'informations à différents rythmes « multi-rate » définissent les modalités du fonctionnement du système, et réagissent aux événements extérieurs (interruptions matérielles). Ils doivent continuellement réagir aux changements de l'environnement du système et calculer certains résultats en temps réel sans délai. Dans ce cas le système est dit « Réactif ».
- Tous les systèmes informatiques ont des contraintes sur les métriques de conception, mais ceux sur les systèmes embarqués peuvent être particulièrement plus durs. Une métrique de conception est une mesure des caractéristiques d'une implémentation, telles que le coût, la taille, les performances et la puissance. Les systèmes embarqués doivent souvent être dimensionnés pour être intégrés sur une seule puce, fonctionner assez vite pour traiter les données en temps réel et doivent limiter la dissipation de puissance lors de l'exécution pour prolonger la durée de vie de la batterie. De plus, le coût de conception et de fabrication doit être le plus faible possible.
- Fonctions dédiées : Un système embarqué exécute habituellement un seul programme ou fonction, à plusieurs reprises.

2.3 Technologie des processeurs embarqués

Un processeur est une unité importante dans un système embarqué. C'est le cœur de ce système [81]. Il peut aller d'un simple microcontrôleur de 8 bits, à un microcontrôleur plus complexe de 32 ou 64 bits, ou même à des multiprocesseurs. Le concepteur de l'embarqué doit sélectionner le choix le plus raisonnable au coût

de l'application et qui peut répondre à toutes les exigences [83]. La technologie du processeur implique l'architecture de l'unité de calcul utilisée pour implémenter la fonctionnalité souhaitée d'un système. Bien que le terme « processeur » soit habituellement associé à des processeurs logiciels programmables, on peut également penser à de nombreux autres systèmes numériques non programmables comme des processeurs utiles pour les systèmes embarqués. Chacun de ces processeurs diffère dans sa spécialisation vers une application particulière, manifestant ainsi des métriques de conception différentes. L'évolution continue de la technologie CMOS et les demandes croissantes de nouveaux produits ont conduit à augmenter non seulement la capacité d'intégration des transistors au sein d'une seule puce mais aussi la fréquence de fonctionnement des composants, tout en respectant la loi de Moore qui stipule que le taux d'intégration des transistors sur une puce et la complexité des circuits intégrés suivrait une courbe exponentielle voir même doubler tous les 18 à 24 mois. Cependant, cette loi a atteint sa limite ou se rapproche de la limite, comme cela est illustré sur la figure 2.2 où la vitesse à laquelle les concepteurs produisant des transistors n'ont pas suivi cette augmentation.

Ainsi, un concepteur doit connaître l'état de l'art des technologies de conception matérielle et logicielle pour être en mesure de réaliser des systèmes embarqués modernes. Toutes ces avancées dans les technologies électroniques numériques en terme de complexité conduisent à des unités de contrôle numériques plus efficaces qui prennent plusieurs formes. Quatre technologies principales sont disponibles [81]:

- GPP (General Purpose Processors) : La fonctionnalité du système est exclusivement réalisée sur le niveau logiciel. Le concepteur d'un GPP réalise un dispositif adapté à une variété d'applications afin de maximiser le nombre des dispositifs développés. Le concepteur d'un système embarqué n'est pas concerné par la conception du processeur, il l'utilise simplement en programmant sa mémoire pour implémenter la fonctionnalité requise. Plusieurs avantages sont obtenus par l'utilisation d'un processeur universel dans un système embarqué. Un temps de conception relativement court, où le concepteur écrit seulement un programme sans passer par la conception du

système numérique. Les performances peuvent être rapides pour des applications à forte intensité de calcul, si elles utilisent un processeur rapide. Ceci est dû aux fonctionnalités d'architectures avancées et à la technologie des circuits intégrés (IC : Integrated Circuit) de pointe. Cependant, il existe également certains inconvénients de conception comme la consommation d'énergie, le coût, la taille physique, la dissipation thermique ainsi qu'une faible performance pour certaines applications [81][83].

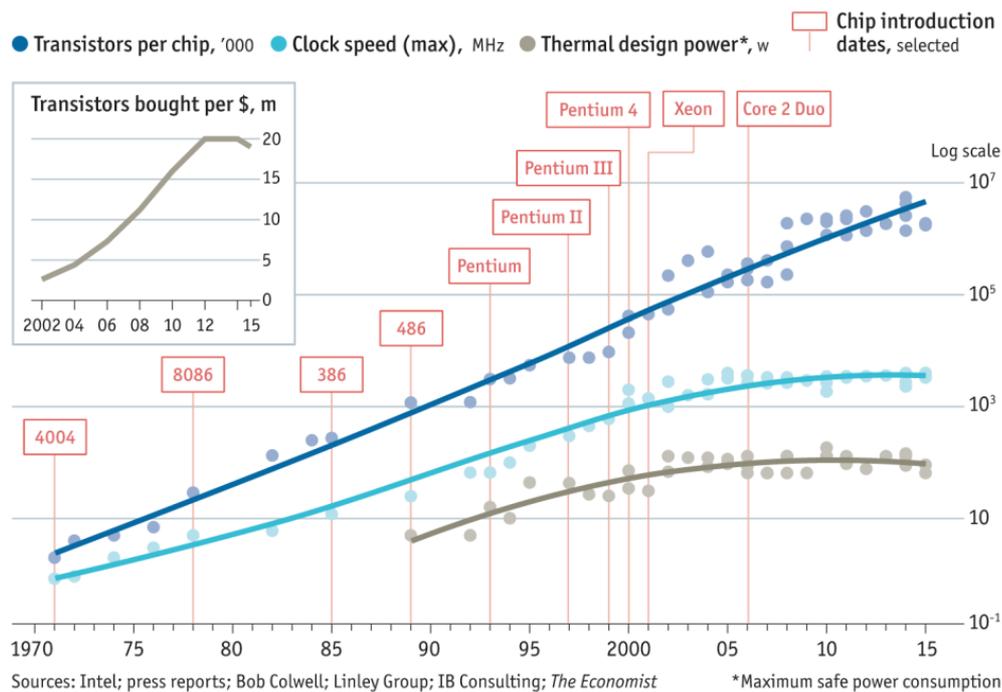


Figure 2.2: Croissance de la technologie des processeurs chez Intel

- ASIC (Application Specific Integrated Circuits) : Un ASIC est fondamentalement un circuit intégré conçu spécialement pour un usage spécial ou une application. Un ASIC diffère d'un produit standard dans la nature de sa disponibilité. La propriété intellectuelle (IP : Intellectual Property), la base de données de conception et le déploiement d'un ASIC sont généralement contrôlés par une seule entité ou entreprise, qui est généralement l'utilisateur final de l'ASIC. Ainsi, un ASIC est de nature exclusive et n'est pas accessible au grand public. Par rapport aux GPP, les systèmes basés sur un ASIC offrent de meilleures performances et une meilleure consommation d'énergie mais au détriment de

la flexibilité et de l'extensibilité. Bien qu'il soit difficile d'utiliser un ASIC pour des tâches autres que celles pour lesquelles ils ont été conçus, il est possible d'utiliser les GPPs pour exécuter les tâches plus générales et moins exigeantes en les assemblant avec les ASICs dans le même système [83][84].

- ASIP (Application Specific Instruction Set Processors) [85]: Ils sont conçus pour une classe particulière d'applications avec des caractéristiques communes telles que le traitement du signal numérique, les télécommunications, le contrôle embarqué, ...etc [86]. Un ASIP est essentiellement un compromis entre les deux extrêmes ; les ASICs qui sont conçus pour faire principalement un travail très spécifique avec des performances élevées, et avec un peu de possibilités pour faire des modifications, et les GPPs qui coûtent beaucoup plus que les ASICs, mais avec une très bonne flexibilité. Grâce à cette flexibilité ainsi qu'au faible coût, l'ASIP est idéal pour être utilisé dans les solutions intégrées et les systèmes sur puce. Cependant, ces processeurs peuvent exiger un coût élevé pour la réalisation du processeur lui-même et pour le développement d'un compilateur, si ces éléments n'existent pas déjà. Les concepteurs utilisant un ASIP développent souvent une grande partie du logiciel en langage assembleur, ce qui rend difficile leur utilisation [84][87].
- FPGA (Field Programmable Gate Arrays) : Ce sont des ICs numériques qui englobent des blocs logiques configurables ainsi que des interconnexions configurables entre ces blocs. Les concepteurs peuvent configurer de tels périphériques pour réaliser une grande variété de circuits. Les FPGAs offrent la possibilité de concevoir des architectures parallèles puissantes qui peuvent réduire considérablement le temps d'exécution de l'algorithme implémentée. Dans le cas des systèmes embarqués, le développement d'une bibliothèque de modules IP basés sur le FPGA est très pratique. Ceci assure la portabilité des modules correspondants entre des cibles différentes. Les FPGAs appartiennent à la classe des architectures reconfigurables qui offrent un compromis entre la flexibilité et l'efficacité. D'autre part, du point de vue technologique, les densités croissantes des FPGAs et la diminution de leurs prix en raison du volume de production élevé permettent l'intégration de systèmes embarqués dans un seul

circuit FPGA. Cela diminue le temps de développement du système pour la partie matérielle et améliore en même temps la possibilité d'utiliser des processeurs dédiés et des circuits ASICs pour fonctionner ensemble dans une application spécifique [86][88].

Tableau 2.1: Comparaison des architectures

	GPP	ASIC	ASIP	FPGA
Performance	Faible	Excellente	Elevée	Très élevée
Flexibilité	Excellente	Mauvaise	Bonne	Très bonne
Conception HW/SW	SW	HW	SW	HW ou HW/SW
Consommation d'énergie	Grande	Très Faible	Moyenne	Faible
Temps de développement	Très faible	Très élevé	Faible	Moyen
Time to market	Très court	Très long	Court	Moyen
Réutilisation	Excellent	Mauvaise	Bonne	Très bonne

2.4 Circuits logiques programmables

Un circuit logique programmable, ou réseau logique programmable, bien connu aussi par son sigle PLD (Programmable Logic Device), est un terme très générique qui regroupe une panoplie de circuits logiques qui peuvent être programmés après leur fabrication [89]. Ce terme (PLD) englobe (Figure 2.3) à la fois les circuits SPLDs (Simple Programmable Logic Devices) ainsi que les circuits CPLDs (Complex Logic Programmable Device). Les PROMs (Programmable Read Only Memory) sont considérés comme les premiers circuits programmables où le bus d'adresses dans ces circuits est utilisé comme entrée et le bus de données comme sortie du circuit logique [90]. Ce type de composant, trouve rapidement ses limitations dans la réalisation des architectures logiques. D'autres types de circuits ont vu le jour tel que les FPLA (Field Programmable Logic Array). La figure suivante montre les différentes familles des circuits logiques programmables.

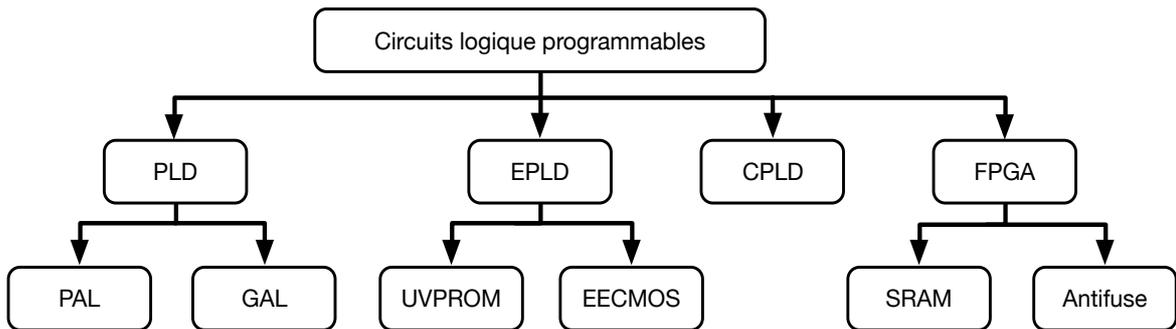


Figure 2.3 : Les différentes familles des circuits logiques programmables

La différence entre ces composants est structurelle. Les CPLDs qui dérivent des PALs (somme de produit : AND-OR), offrent une solution à une seule puce avec de faibles délais de broche à broche, même dans le cas des fonctions à plusieurs entrées. Une fois programmée, la conception peut être verrouillée et sécurisée. La plupart des architectures avec des CPLDs sont très similaires. La complexité limitée (<500 flip-flops) signifie que la plupart des CPLDs sont utilisés pour des fonctions logiques de liaison. De plus, leur consommation élevée en énergie interdit leur utilisation dans les équipements fonctionnant sur batterie [90].

Les FPGAs offrent une complexité beaucoup plus élevée et leur consommation d'énergie statique est faible, bien qu'elle augmente fortement dans les nouvelles familles. Comme le bitstream de configuration doit être rechargé chaque fois que le circuit est réalimenté, la sécurité du système conçu est affaiblie, mais les avantages et les possibilités de la reconfiguration dynamique et le prototypage rapide des circuits numériques à grande complexité constituent un avantage important. Les FPGAs offrent une plus grande souplesse et des fonctionnalités plus avancées que les CPLDs [90] : gestion d'horloge, RAM sur puce, fonctions DSP (multiplicateurs) et même des microprocesseurs sur puce.

2.4.1 SPLDs et CPLDs

Les premiers CI (Circuit Intégré) programmables sont appelés circuits logiques programmables (PLD). Les composants originaux, qui ont commencé à arriver sur le marché en 1970 sous la forme des PROMs, étaient assez simples. Et ce n'est

qu'à la fin des années 1970, que des versions beaucoup plus complexes sont devenues disponibles. Afin de les distinguer de leurs ancêtres moins sophistiqués, qui trouvent encore aujourd'hui leur utilité, ces nouveaux dispositifs ont été appelés PLDs complexes ou CPLDs [89]. Certains utilisateurs voient les termes PLD et SPLD comme des synonymes, tandis que d'autres considèrent les PLDs comme étant un ensemble qui englobe à la fois les SPLDs et les CPLDs [90].

Avec leur architecture basique en portes logiques AND/OR programmables, les PALs sont considérés comme des circuits flexibles qui peuvent implémenter des équations logiques sous les différentes formes booléennes en somme de produits ou produit de sommes. Certaines améliorations sont apportées aux SPLDs comme les Entrées/Sorties programmables et bidirectionnelles et plusieurs configurations des registres et des horloges [91].

Un grand avantage des PLDs est qu'ils peuvent remplacer les circuits intégrés numériques de petite et moyenne échelle pour une densité d'assemblage plus élevée. Un PLD pourrait remplacer des dizaines de circuits intégrés de 200 à 500 portes logiques. D'autres avantages pour les SPLDs est qu'ils réduisent la consommation d'énergie, offrent des performances plus rapides (car ils réduisent les interconnexions entre les circuits) et une fiabilité plus élevée. Les SPLDs sont disponibles en technologie bipolaire et CMOS (Complementary Metal Oxide Semiconductor).

Des circuits de densité supérieure font leur apparition sur le marché. Ils regroupent plusieurs SPLDs sur la même puce pour composer un CPLD, et ce dans le but d'augmenter les performances et les capacités de l'architecture. Généralement les CPLDs rassemble des macro-cellules qui communiquent entre-elles en utilisant des signaux internes pouvant être routés à l'aide d'une matrice d'interconnexion programmable (Figure 2.4) [91]. Réservés à des fonctions simples ou moyennement complexes, ces circuits sont rapides (jusqu'à 200 MHz) et leurs caractéristiques temporelles sont pratiquement indépendantes de la fonction réalisée.

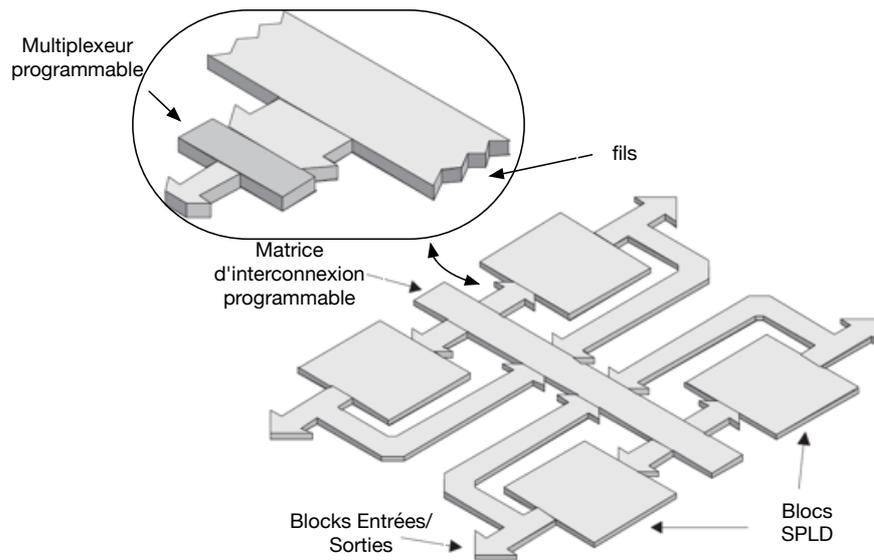


Figure 2.4 : Une structure CPLD générique

2.4.2 FPGAs

Les circuits FPGAs ont été introduits comme une simple technologie des circuits logiques [90]. Ils ont connu une croissance rapide et sont devenus un moyen d'implémentation populaire pour les circuits numériques. L'évolution de la technologie a considérablement amélioré la capacité logique des FPGAs et les a rendus comme une alternative d'implémentation pour des conceptions plus grandes et complexes [91]. En outre, la nature programmable de leurs ressources logiques et de routage a un grand effet sur la taille de la surface, la vitesse et la consommation d'énergie du circuit. Les FPGAs peuvent être reprogrammés électroniquement selon les exigences d'application ou de fonctionnalité souhaitée pour devenir pratiquement n'importe quel type de circuit ou système numérique [92]. Cette fonctionnalité distingue les FPGAs des ASICs, qui eux sont fabriqués sur mesure pour des tâches de conception spécifique. Cette approche permet de corriger les erreurs de conceptions détectées au niveau final du développement, éventuellement par une simple reprogrammation de l'FPGA, permettant ainsi de modifier l'inter-connectivité des composants selon les besoins. Bien que cette approche introduise des retards supplémentaires en raison de l'interconnexion programmable, elle évite une reconception coûteuse [93].

La nature souple des FPGAs les rend considérablement plus grands, plus lents et plus consommateurs d'énergie que leurs homologues ASICs. Ces inconvénients résultent en grande partie de l'interconnexion de routage programmable qui comprend près de 90% de la superficie totale des FPGAs. Cependant, malgré ces inconvénients, les FPGAs offrent une bonne solution pour l'implémentation des systèmes numériques en raison de leur faible coût et du délai de mise sur le marché plus rapide par rapport aux ASIC, qui nécessitent en générale beaucoup de temps et d'argent pour obtenir un premier prototype.

Les FPGAs comprennent dans leur structure de base, les composants suivants (Figure 2.5)[94]:

1. Blocs logiques configurables (CLB : Configurable Logic Block) qui implémentent des fonctions logiques à travers des LUTs (LookUp Table). Ces blocs sont placés dans une grille bidimensionnelle et sont interconnectés par des réseaux d'interconnexion programmables.
2. Les blocs d'Entrées/Sorties qui sont placés sur la périphérie de la grille et sont connectés à des blocs logiques via le réseau d'interconnexion programmable.
3. Le réseau d'interconnexions configurables entre ces différents blocs.

L'architecture interne des FPGAs actuels comporte, dans sa structure, des blocs additionnels tels que les registres pour stocker les résultats, des blocs mémoires RAM à haute vitesse qui permettent d'implanter des mémoires adressables, des ressources spécifiques comme les blocs génériques DSP dédiés à la réalisation facile de nombreuses opérations de traitement, ainsi que des processeurs hardware intégrés [95].

Il existe un certain nombre de technologies de configuration par la programmation qui ont été utilisées pour les architectures reconfigurables comme les FPGAs. Ces technologies ont des caractéristiques différentes qui ont à leur tour un effet significatif sur l'architecture reconfigurable. Certaines des technologies bien

connues incluent la mémoire statique, flash et anti-fusible [93].

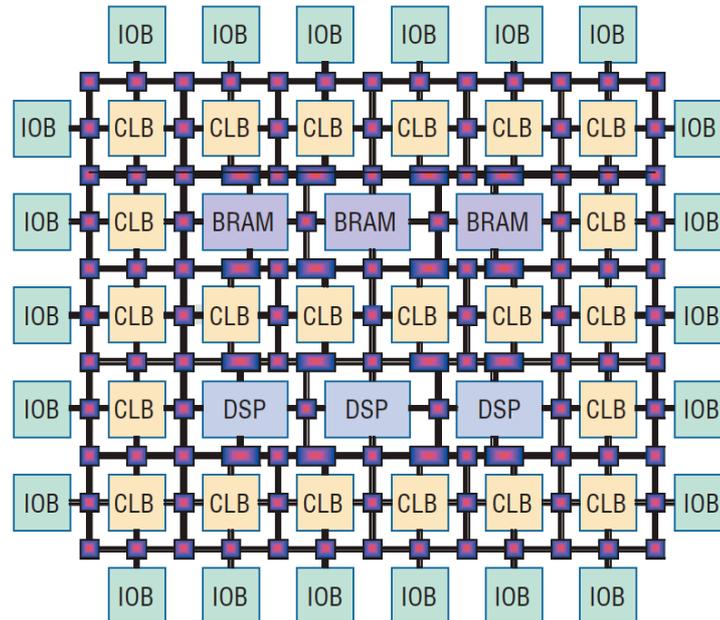


Figure 2.5 : Structure interne de l’FPGA basée sur l’architecture de Xilinx

2.4.2.1 Technologie de programmation SRAM

Il existe deux versions principales de périphériques RAM à semi-conducteurs : RAM dynamique (DRAM) et RAM statique (SRAM). Dans le cas des DRAM, chaque cellule est formée d'une paire de transistors-condensateurs. Le terme "dynamique" est utilisé car le condensateur perd sa charge dans le temps, de sorte que chaque cellule doit être périodiquement rechargée si elle doit conserver ses données. Cette opération, connue sous le nom de rafraîchissement, est un peu complexe et nécessite une quantité substantielle de circuits supplémentaires. Lorsque le coût de ce circuit de rafraîchissement est amorti sur des dizaines de millions de bits dans un dispositif de mémoire DRAM, cette approche devient très rentable. Cependant, la technologie DRAM présente peu d'intérêt pour la logique programmable [93].

Par analogie, le terme "statique" associé à la SRAM est utilisé, car une fois qu'une valeur a été chargée dans une cellule SRAM, elle reste inchangée à moins qu'elle ne soit modifiée spécifiquement ou jusqu'à ce que le circuit soit mis hors tension

[91].

La technologie de programmation basée sur la mémoire statique (SRAM) est devenue l'approche dominante pour les FPGAs, en raison de sa ré-programmabilité et de l'utilisation d'une technologie CMOS standard. Ceci permet d'augmenter le taux d'intégration sur puce et donc une vitesse plus élevée et une consommation d'énergie dynamique plus faible avec une surface plus petite. Actuellement, la plupart des fabricants tels que Lattice et Xilinx adoptent cette technologie dans leurs circuits. Ces derniers utilisent des cellules de mémoire statique qui sont réparties sur tout l'FPGA pour assurer la configurabilité. Les cellules SRAM sont principalement utilisées pour les buts suivants [90]:

- Programmer le réseau d'interconnexions des FPGAs qui sont généralement pilotés par des multiplexeurs.
- Programmer des CLBs qui sont utilisés pour implémenter des fonctions logiques.

Il existe cependant un certain nombre d'inconvénients associés à la technologie de programmation SRAM. Par exemple, une cellule SRAM nécessite 6 transistors, ce qui rend l'utilisation de cette technologie coûteuse en termes de superficie par rapport à d'autres technologies de programmation. D'autres part, les cellules SRAM sont de nature volatile et des circuits externes sont nécessaires pour stocker en permanence les données de configuration. Ces circuits externes s'ajoutent au coût et à la surface de fonctionnement des FPGAs [96].

2.4.2.2 Technologie de programmation Flash

Une alternative à la technologie de programmation SRAM est l'utilisation de la technologie de programmation basée sur une mémoire flash ou EEPROM. Cette technologie offre plusieurs avantages. Par exemple, elle est de nature non volatile et est également plus efficace que la technologie de programmation SRAM. Cependant, elle a également ses propres inconvénients. Contrairement à la

technologie basée sur la SRAM, les composants à mémoire flash ne peuvent pas être reconfigurés un grand nombre de fois. De plus, la technologie flash utilise une technologie CMOS non standard [96].

2.4.2.3 Technologie de programmation anti-fusible

La technologie de programmation anti-fusible est une alternative aux technologies SRAM et flash, dans laquelle chaque chemin configurable a une liaison associée appelée anti-fusible [96].

Un anti-fusible représente au début une colonne microscopique de silicium amorphe (non-cristallin) reliant deux pistes métalliques. Dans son état non programmé, le silicium amorphe agit comme un isolant avec une très haute résistance supérieure à un milliard d'ohms. Il peut être considéré comme un circuit ouvert, c'est l'état initial de l'FPGA avant qu'il ne soit programmé. Les anti-fusibles peuvent être programmés en appliquant des impulsions de tension et de courant relativement élevées à ses entrées, en convertissant le silicium amorphe isolant en silicium polycristallin conducteur [93].

Le principal avantage de la technologie de programmation anti-fusible est sa faible surface. En outre, cette technologie est non volatile par défaut. Il existe toutefois d'importants inconvénients associés à cette technologie de programmation. Par exemple, cette dernière n'utilise pas la technologie CMOS standard. En outre, les circuits basés sur cette technologie ne peuvent pas être reprogrammés [93].

2.5 Systèmes sur Puce

Un système sur puce (SoC : System on Chip) est défini comme un circuit conçu et fabriqué pour un usage spécifique et pour une utilisation exclusive [84]. Le terme SoC désigne aussi l'intégration d'un système complet, avec de nombreux composants, comme un ou plusieurs processeurs, un bus, des circuits analogiques, une mémoire embarquée, et des éléments logiques programmable, le tout au sein d'une seule puce. D'une manière simple, nous définissons un SoC comme un circuit

intégré, conçu par assemblage de plusieurs conceptions VLSI « Very-Large-Scale Integration » autonomes afin de fournir toutes les fonctionnalités d'une application [97].

Un SoC devient capable d'exécuter des logiciels intégrés dès lors qu'on intègre dans son architecture au moins un processeur. Certains considèrent qu'un SoC n'est qu'un ASIC. En comparaison, un ASIC n'inclut pas nécessairement un processeur et dans ce cas, un SoC peut être considéré comme une forme particulière d'un ASIC [98].

Grâce à l'amélioration des technologies des semi-conducteurs, à la conception conjointe matérielle/logicielle "hardware/software codesign" et à l'utilisation des blocs de propriété intellectuelle IP (Intellectual Property), la méthodologie de conception des SoCs a considérablement évolué et de nombreux obstacles ont été résolus, comme la densité d'intégration, la réduction des coûts de conception et l'amélioration des niveaux de flexibilité [98].

L'architecture d'un SoC peut englober des combinaisons des IPs qui peuvent être une fonction de base ou bien un bloc plus complexe telles que des microprocesseurs, des blocs mémoires SRAM/DRAM, des contrôleurs audio et vidéo, des convertisseurs A/N et N/A, des contrôleurs graphiques 2D et 3D, des fonctions DSP et des blocs d'interface et de communications tels que l'UART, PCI, et USB, ...etc. La Figure 2.6 [98] représente l'architecture interne d'un système sur puce typique.

Ces IPs sont généralement disponibles soit dans une forme de description de niveau de transistor, ou bien décrite par un langage de description matériel de haut niveau (HDL) tel que le VHDL ou le Verilog. Leur flexibilité d'utilisation dépend de la forme sous laquelle ils sont disponibles. Un composant IP peut apparaître sous différentes formes [99]:

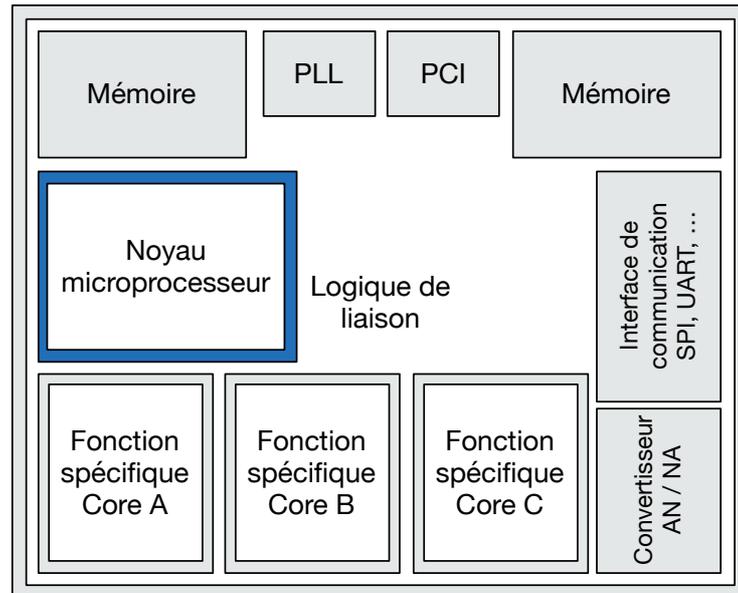


Figure 2.6 : Architecture interne d'un SoC typique

- **IP Logiciel (soft-core)** : Ce sont des blocs réutilisables sous la forme d'une description RTL (Register Transfer Level) synthétisable ou une « netlist » d'une librairie générique. Cela implique que l'utilisateur de l'IP logiciel est le seul responsable de l'implémentation, ainsi que de l'optimisation du circuit final. Les principaux avantages de l'utilisation de l'IP logiciel est sa portabilité et sa flexibilité. Son inconvénient est qu'il ne peut pas être prédictif en termes de superficie, de puissance consommée et de temps de réponse.
- **IP Matériel (hard-core)** : Ce sont des blocs réutilisables qui ont été optimisés pour améliorer les performances, la puissance consommée et la taille. Ils sont mappés par une technologie spécifique ; dans ce cas, la description du bloc IP est au niveau porte logique. Ceux-ci existent comme un netlist complètement placé et routé, et comme un circuit fixe. Contrairement à l'IP logiciel, un IP matériel possède une plus grande prédictibilité mais il est moins flexible.
- **IP Firm (firmcore)** : Il s'agit de blocs réutilisables qui ont été optimisés sur le plan structurel et topologique pour améliorer la performance et la taille. Ceux-ci existent sous forme de code HDL synthétisé et réalisé pour une technologie cible ou comme une netlist d'une librairie générique. Un IP firmcore offre plus

de flexibilité que le hard-core, et plus de prédictibilité en termes de performance et de taille que le soft-core.

Le cœur du processeur représente un élément critique dans un SoC. Il doit fournir le maximum de performance, être le plus flexible, mais toujours sur une surface la plus réduite possible. Selon plusieurs études réalisées ces dernières années, les processeurs destinés aux SoCs peuvent être divisés en deux catégories : soit sous forme des macros spécialisées, dites processeurs hard-cores (PowerPC de Xilinx et ARM9 d'Altera), soit sous forme des modules synthétisables, dites processeurs soft-cores (PicoBlaze, MicroBlaze, NIOS-II, etc.).

- **Processeur hard-core**

C'est un processeur non synthétisable, décrit au niveau physique, donc il ne peut pas être reconfiguré et tous ses composants sont déjà fixés par une disposition personnalisée utilisant des conceptions VLSI et qui sont intégrés dans un FPGA avec d'autres ressources internes. Le cœur du processeur hard offre des avantages de traitement et de communication rapides, et ses performances en surface, consommation, et délais sont connues aux préalables, et ne varient pas d'un système à l'autre. Toutefois, le processeur hard-core souffre d'une trop faible flexibilité [100].

- **Processeur soft-core**

C'est un processeur synthétisable, décrit par une approche logicielle, généralement avec un langage de description matériel (VHDL ou bien Verilog), et qu'il peut être entièrement synthétisé et implémenté sur un circuit logique programmable tel que l'FPGA. L'avantage de ces processeurs réside dans leur capacité d'être paramétré et la possibilité de choisir les options de configuration. Par conséquent, le concepteur peut choisir les fonctionnalités du processeur en fonction des besoins d'une application cible spécifique. Le processeur soft-core permet d'avoir une meilleure flexibilité par rapport au processeur hard-core et offre un compromis entre les performances et le coût, et permet ainsi l'accélération du cycle de conception,

mais ses performances restent toujours en dessous de celles des processeurs hard-core [101].

Le tableau 2.2 présente une comparaison des caractéristiques de plusieurs processeurs soft-core. La première colonne représente les caractéristiques à travers lesquelles les processeurs sont comparés. Les colonnes suivantes montrent les fonctionnalités disponibles à l'intérieur de chaque processeur soft-core. Comme le montre ce tableau, les processeurs soft-core Nios II et MicroBlaze sont ciblés et optimisés principalement pour une implémentation sur FPGA. En revanche, les trois autres cœurs ne sont pas optimisés pour une technologie cible spécifique.

Tableau 2.2 : Comparaison des processeurs soft-core [101]

Catégorie	Nios II	MicroBlaze	Xtensa XL	OpenRISC 1200	LEON3
Fréquence max MHz	200 (FPGA)	200 (FPGA)	350 (ASIC)	300 (ASIC)	400/125 (ASIC/FPGA)
Technologie ASIC / FPGA	Startix / Startix II	Virtex-4	0.13 μm	0.18 μm	0.13 μm /Not given
DMIPS	150 DMIPS	166 DMIPS	-	250 DMIPS	85 DMIPS
ISA	32bits RISC	32 bits RISC	32 bits RISC	32 bits RISC	32 or 64bits RISC
Mémoire cache	Jusqu'à 64 KB	Jusqu'à 64 KB	Jusqu'à 32 KB	Jusqu'à 64 KB	Jusqu'à 256 KB
Pipeline	6 étages	3 étages	5 étages	5 étages	7 étages
Instructions	Jusqu'à 256 Instruction	None ??	Illimité	Limite non spécifiée	None ??
Taille des registres	32 bits	32 bits	32 où 64 bits	32 bits	2 à 32 bits
Implémentation	FPGA	FPGA	FPGA, ASIC	FPGA, ASIC	FPGA, ASIC
Zone occupé	700-1800 Les	1269 LUTs	0.26 mm^2	N/A	N/A

2.6 Co-design : Conception conjointe matérielle/logicielle

La conception du système est l'opération de mise en œuvre d'une fonctionnalité

souhaitée à l'aide d'un ensemble de composants matériels ou logiciels. Le mot système fait référence à tout dispositif fonctionnel implanté dans le matériel ou le logiciel ou bien les combinaisons des deux. Lorsqu'il s'agit d'une combinaison du matériel et du logiciel, nous l'appelons une conception conjointe matérielle/logicielle ou bien tout simplement co-design.

Le terme de co-design est apparu au début des années 1990 pour décrire une confrontation de problèmes dans la conception des circuits intégrés (CI). Les microprocesseurs étaient utilisés depuis plus d'une décennie. Une classe de concepteurs qui étaient largement séparés des concepteurs des circuits intégrés ont intégré des microprocesseurs avec des composants matériels standards sur une carte. Une grande partie du code était en langage assembleur. Il était clair à ce moment-là que la conception de systèmes basés sur un microprocesseur deviendrait une discipline de conception importante pour les concepteurs de circuits intégrés. Des microprocesseurs de 16 bits et de 32 bits avaient déjà été utilisés dans les conceptions des circuits au niveau matériel et il était évident que la loi de Moore conduirait finalement à des puces assez grandes pour inclure à la fois un processeur et d'autres sous-systèmes. Cela soulève deux catégories de problèmes : les méthodologies de conception du système devraient traiter de grandes CPU préconçues, et les logiciels devraient être traités comme une composante de première classe dans la conception des puces [102].

L'introduction récente de technologies FPGA a fondu la distinction entre le matériel et le logiciel. Traditionnellement, un circuit matériel était configuré au moment de la fabrication. Les fonctions d'un circuit matériel pourraient être choisies par l'exécution d'un programme. Alors que le programme pouvait être modifié même au moment de l'exécution, et la structure du matériel étant invariante[103].

Avec la technologie programmable sur site (field-programmable), il est possible de configurer l'interconnexion des portes logiques des circuits après la fabrication. Cette flexibilité ouvre de nouvelles applications de circuits numériques et de nouveaux problèmes de co-design matériel/logiciel se posent. Par exemple, un ou plusieurs circuits FPGAs peuvent être configurés pour implémenter une fonction logicielle

spécifique ayant de meilleures performances que l'exécution du code correspondant sur un microprocesseur [88] [102].

Par la suite, le FPGA peut être reprogrammé pour effectuer une autre fonction spécifique sans changer le matériel sous-jacent. Ainsi, d'un point de vue utilisateur, une plateforme reprogrammable peut exécuter une fonction qui ne peut pas être distinguée de celle d'un processeur. Néanmoins les mécanismes de programmation et la vision du programmeur du matériel sont très différents.

Le grand intérêt de la conception conjointe matérielle/logicielle est dû à l'introduction d'outils de conception assistée par ordinateur (CAO) [88](par exemple, les simulateurs commerciaux) dans l'espoir que des solutions à d'autres problèmes de conception soient pris en charge par ces outils qui augmentent ainsi la qualité potentielle et raccourcissent le temps de développement des produits électroniques.

2.7 Flot de conception conjointe matérielle/logicielle

Le processus global du co-design commence par l'identification des exigences du système. Ce sont les fonctions requises, telles que la performance, la puissance, le coût, la fiabilité et le temps de développement du système. Ces exigences constituent les spécifications préliminaires souvent produites par les équipes de développement et les professionnels du marketing.

En ce qui concerne les étapes de conception, la figure 2.7 montre un flot de méthodologie générique de co-design. Des modèles de simulation sont créés, analysés et validés à chaque étape afin de réduire le risque d'erreurs. Le flot de conception peut être modifié et certaines étapes peuvent être supprimées pour des conceptions spécifiques.

Les étapes formant le flot de conception incluent le partitionnement, la planification et la synthèse de communication. Après ces étapes, un algorithme de haut niveau et un modèle de simulation pour le système global sont créés en utilisant le langage C ou C ++.

D'abord, une spécification au niveau du système de l'application à réaliser est proposée, afin de décrire le comportement fonctionnel du système ainsi que son modèle algorithmique et les spécifications requises par la co-simulation. A la fin de cet étape, l'architecture du système subit un partitionnement matériel/logiciel pour déterminer quelles fonctions doivent être effectuées par le matériel et celles qui doivent être réalisées par les applications logicielles. Pour choisir la meilleure répartition entre les parties logicielles et les parties matérielles, plusieurs tentatives de simulation peuvent être effectuées. Le partitionnement des sous-systèmes logiciels et matériels est actuellement un processus manuel nécessitant de l'expérience et un compromis coût/performance. Une fois les partitions matérielles et logicielles déterminées, un modèle comportemental du matériel est créé avec un prototype fonctionnel du logiciel. La co-simulation du matériel et du logiciel permet d'affiner ces composants et de développer un modèle exécutable avec des spécifications entièrement fonctionnelles. Ces améliorations se poursuivent tout au long de la phase de conception [98].

Après le partitionnement logiciel/matériel, l'étape de planification aura lieu afin de déterminer l'ordre dans lequel une tâche doit s'exécuter sur un élément de traitement (tel qu'un CPU), en tenant compte de l'ensemble des tâches ainsi que de l'ensemble des contraintes de performances.

Dans la planification, les principales considérations sont : (1) l'ordre partiel imposé par les dépendances dans la fonctionnalité ; (2) la minimisation de la surcharge de synchronisation entre les éléments de traitement ; (3) la réduction de la surcharge de changement du contexte dans les éléments de traitement.

L'étape de synthèse appelée aussi d'implémentation aura lieu. Dans cette étape, nous obtenons une description au niveau RTL pour la partie matérielle et un code source (e.g. C ou C++) pour la partie logicielle du système. Bien évidemment, la vérification et la validation de la fonctionnalité du modèle ainsi généré devraient être faites. A ce stade de conception, nous pouvons effectuer des analyses de performances de l'architecture au niveau cycle et au niveau bit à travers des co-simulations.

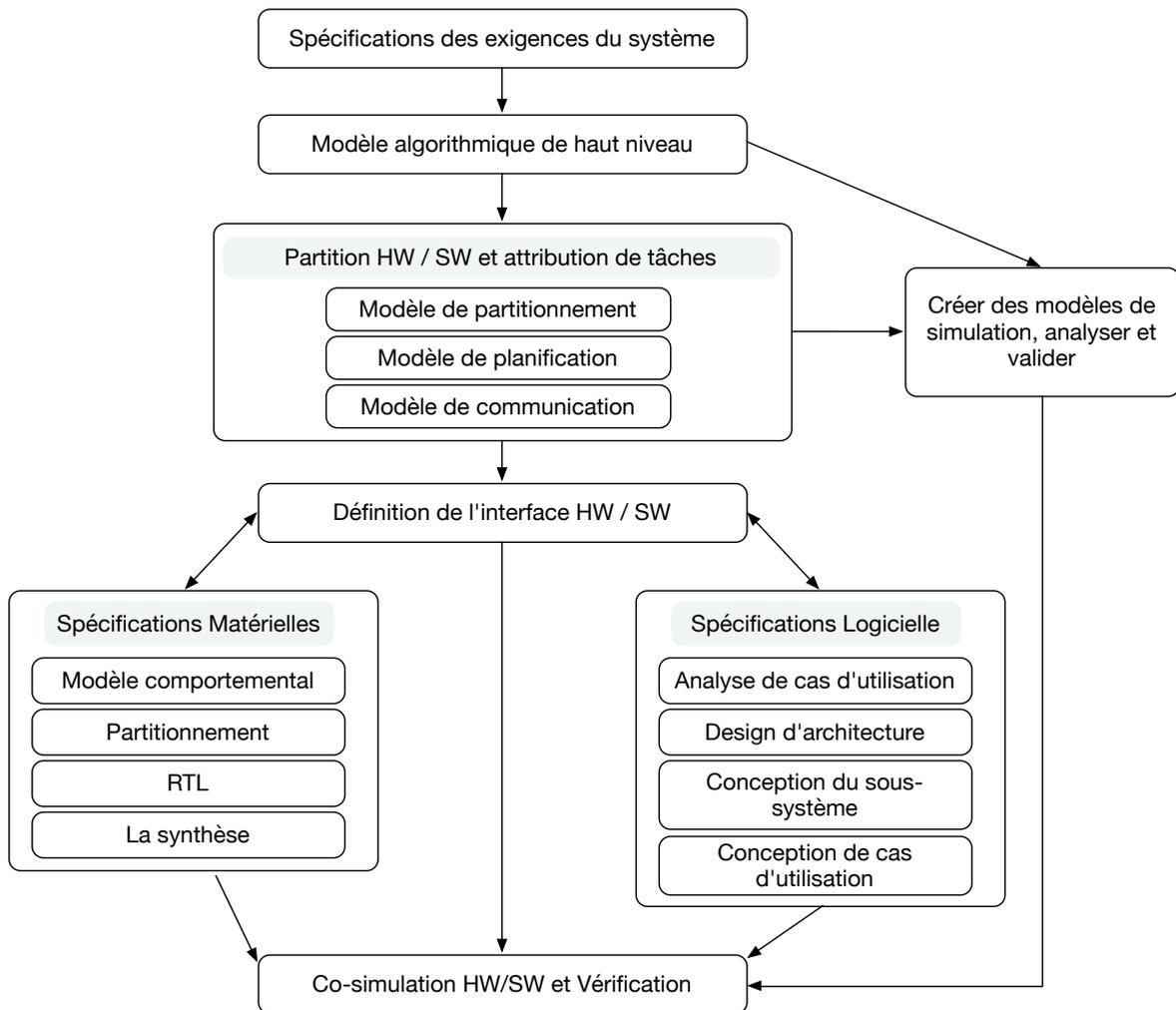


Figure 2.7 : Méthodologie de conception conjointe matérielle/logicielle

L'étape qui suit la planification, sera donc la synthèse dite aussi implémentation. A l'issue de cette étape, une description au niveau RTL est obtenue pour la partie matérielle et un code source généralement en C ou bien C++ pour la partie logicielle. A la fin de cette étape, si le comportement matériel est affecté à un processeur standard, il sera chargé dans le compilateur de ce processeur. Ce compilateur traduit la description de conception en code machine pour le processeur cible. S'il doit être mappé dans un ASIC, un outil de synthèse de haut niveau peut le synthétiser. Le synthétiseur de haut niveau traduit le modèle de conception comportementale en un netlist des composants d'une librairie RTL. La co-simulation à ce stade de la conception, va permettre une analyse de performances de l'architecture au niveau cycle et au niveau bit (occupation de

surface).

La dernière étape du flot de conception est la génération d'une conception RTL à partir du modèle d'interface et du travail de synthèse d'interface. Les interfaces synthétisées seront placées et routées sur le plan de la puce. Pour la partie logicielle du système, elle sera compilée pour générer une image hexadécimale. Enfin, nous pouvons obtenir les performances en surface et en consommation d'énergie à travers des co-simulations.

2.8 Flot de conception FPGA

La conception d'un SoC sur FPGA nécessite l'utilisation d'outils de développement logiciels et matériels appropriés fournis par les fabricants des FPGAs. La figure 2.8 montre le flux de conception standard pour le développement d'applications SoC sur un FPGA de Xilinx. Xilinx dans son pack d'outils fournit les différents soft permettant la création des systèmes embarqués sur puce. Parmi ces softs, on a le Vivado Design Suite qui est un outil complet pour la création des plates-formes matérielles, la vérification et la création du fichier bitstream pour la programmation des FPGA suivant l'application ciblée. Le SDK (Software Development Kit) pour développer des applications définies par l'utilisateur.

La conception est composée de deux procédures principales : Un flot de conception matériel et un flot de conception logiciel.

Le premier flot inclut la conception matérielle et les outils de vérification (éditeur VHDL/Verilog, le synthétiseur, le placement/routage et les outils de mise en œuvre et de simulation). Le deuxième flux offre une interface conviviale qui permet au concepteur de personnaliser le processeur pour un projet spécifique.

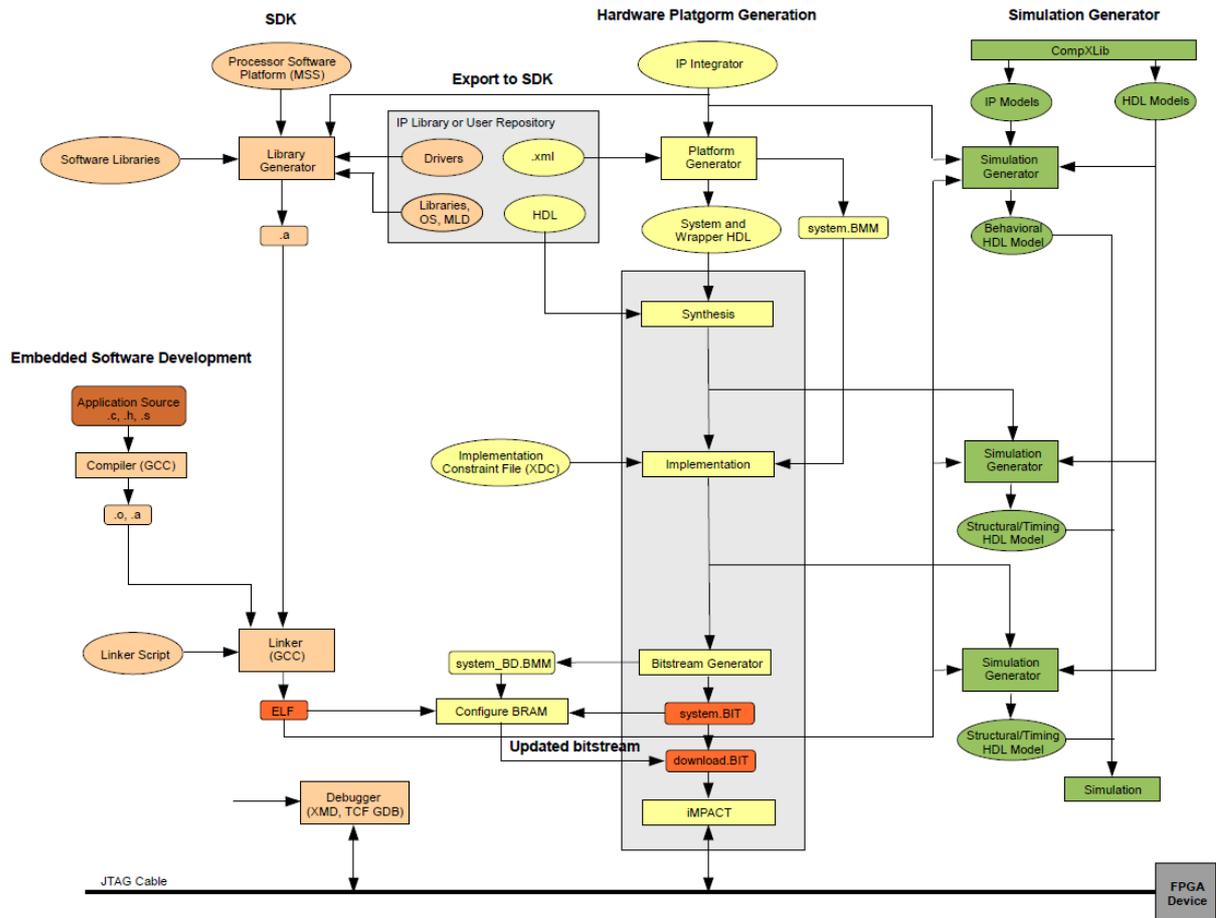


Figure 2.8 : Flux de conception des systèmes embarqués typiques

Après la création et la configuration du processeur soft-core et, ses périphériques par l'utilisation de l'outil Vivado IP integrator, le cœur du processeur est généré sous la forme d'un fichier netlist. Ensuite, ce fichier peut être intégré dans le flux de conception matériel pour la synthèse et le placement/routage. Le fichier bitstream résultant est utilisé par la suite pour la configuration de L'FPGA. Ensuite, la description de la plateforme matérielle est exportée vers le SDK. Le fichier exporté contient toutes les informations nécessaires requises par le SDK pour le développement de logiciels et les travaux de débogage sur la plate-forme matérielle conçue. Le programme qui sera intégré sur le processeur soft-core peut être compilé avec les fichiers de bibliothèques associés et les fichiers d'en-tête C. Maintenant, la conception proposée est prête à être implémentée. La dernière étape sera d'initialiser le fichier bitstream (.bit) avec le fichier exécutable ELF (.elf) approprié et de le charger sur le circuit FPGA cible.

2.9 Intérêt de l'approche SoC pour les nœuds de réseaux de capteurs sans fil

Sur la base de ce qui précède, la conception des nœuds capteurs pour les applications industrielles devrait surmonter plusieurs défis tels que le temps de traitement qui devrait être réduit sans influencer la qualité de service, étant donné que les systèmes de contrôle dans les nouvelles applications industrielles sont généralement basés sur des algorithmes sophistiqués qui ont besoin de beaucoup de ressources informatiques. L'objectif principal de ce travail est la conception et l'implémentation des nœuds sur puce d'un réseau de capteurs sans fil pour un système de surveillance des vibrations. Ces nœuds utilisent un processeur embarqué et un matériel reconfigurable personnalisé.

L'adoption d'une conception basée sur un SoC est justifiée parce que c'est une approche prometteuse pour améliorer les performances des nœuds en particulier pour les applications industrielles rigoureuses et offre une bonne solution pour le traitement du signal en temps réel.

2.10 Méthodologie de conception conjointe proposée pour les nœuds capteurs sans fil

Le développement d'un nœud de capteur sans fil commence par la description des spécifications de l'ensemble de l'application. Cela inclut la définition du système physique utilisé et les paramètres de contrôle. La spécification du système physique consiste à choisir le type de processeur, le module de communication, les capteurs nécessaires ainsi que l'interface analogique numérique (ADC). Les conditions d'environnement (milieu fermé ou ouvert) doivent également être prises en compte car elles affectent les communications entre les différents nœuds du réseau.

Dans ce travail, les différents nœuds du système sont implémentés sur des plateformes FPGA basées sur un processeur soft-core MicroBlaze de Xilinx. Le choix d'une conception conjointe matérielle/logicielle va nous permettre non seulement l'amélioration de l'efficacité du système mais aussi sa fiabilité.

L'étape suivante consiste à développer, tester et valider les IPscore personnalisés nécessaires dans l'architecture globale du système, qui n'existent pas sur la bibliothèque des IPs core de Xilinx. Dans notre cas, ce sont les blocs de cryptage et de décryptage AES (Advanced Encryption Standard).

Une fois que l'architecture globale est implémentée et validée, le concepteur peut entreprendre la phase du développement architectural. Il consiste à optimiser l'allocation des ressources hardware et software lors de l'implémentation. Cela nécessite une méthode d'exploration de l'espace architectural de conception appropriée basée sur deux étapes principales : l'estimation de performance et le partitionnement hardware/software. La première étape permet d'estimer la surface, le temps de réponse et la taille de la mémoire pour chaque module de l'architecture. Ensuite, le partitionnement hardware/software est effectué. L'objectif est de minimiser l'espace, le temps et la taille de la mémoire par rapport aux contraintes fonctionnelles et aux contraintes architecturales (ressources disponibles). Plus de détails sont fournis au chapitre 3.

Une fois que le partitionnement hardware/software est effectué, le concepteur commence l'étape d'intégration hardware/software de la solution optimale choisie. Il consiste à développer le code VHDL des différents modules à implémenter dans le hardware (généralement générer automatiquement par Vivado) et le développement du code C de ceux qui seront implémentés sur le software. La communication entre les deux parties est autorisée en utilisant les interfaces de communication hardware/software. L'intégration de toutes ces parties représente l'étape suivante. Selon le flux de conception donné à la figure 2.9 ci-contre, le processus physique peut être effectué et le Bitstream téléchargé. Enfin, la validation expérimentale de la solution choisie peut être réalisée.

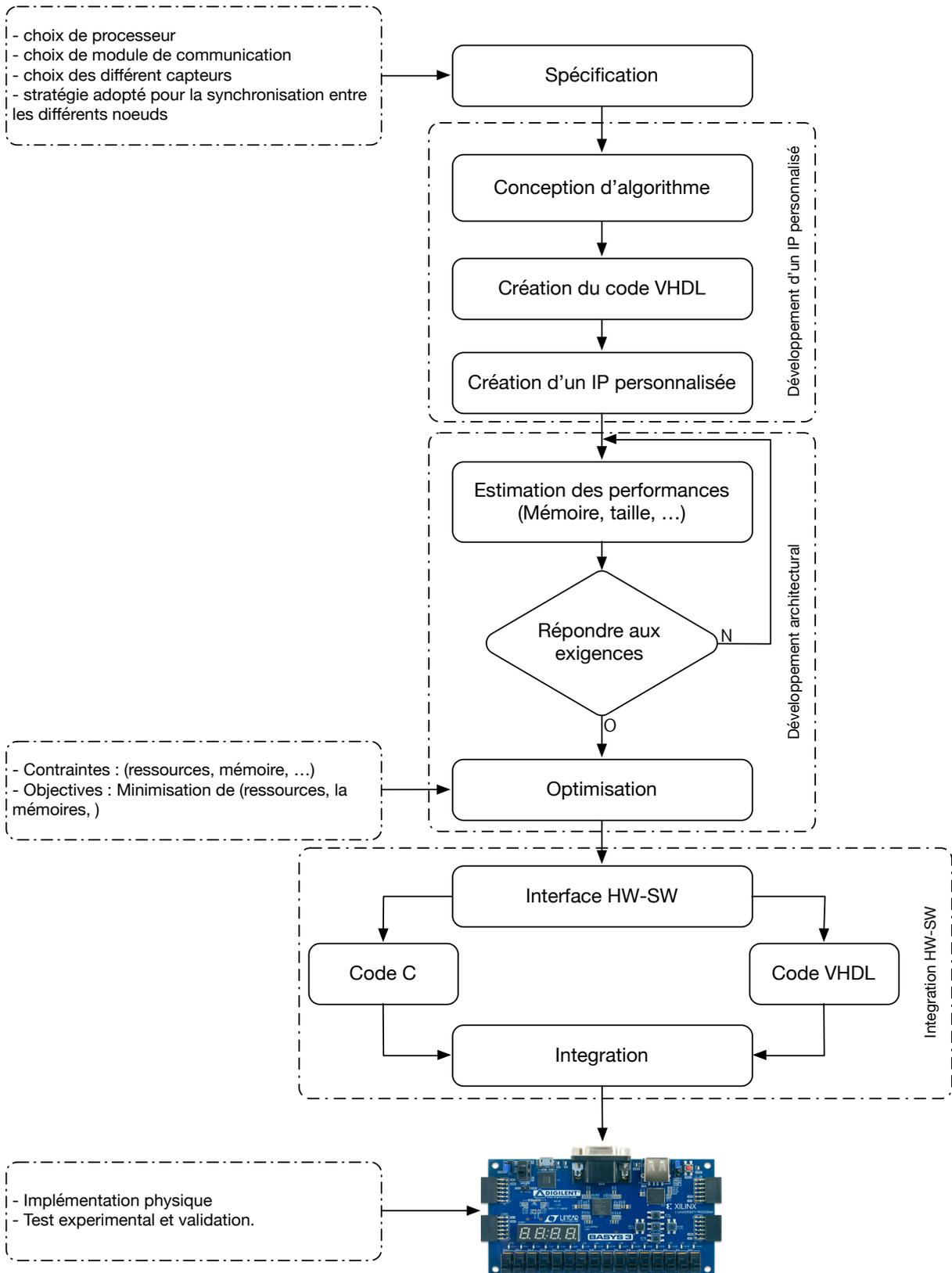


Figure 2.9 : Conception conjointe d'un nœud de capteur sans fil

2.11 Conclusion

Dans ce chapitre, nous avons présenté une vue d'ensemble sur les systèmes embarqués ainsi qu'une brève présentation des processeurs disponibles (GPP, ASIP, ASICs, FPGA) utilisés dans les systèmes embarqués. Une description des circuits logiques programmables et plus précisément les FPGA ont été étudiés. Ensuite, l'approche SoC ainsi que la conception conjointe hardware/software ont été introduites. Enfin, la méthodologie de conception conjointe proposée pour les nœuds de capteurs sans fil a été étudiée. Cette méthode couvre toute la chaîne de développement allant des spécifications à la validation pratique basée sur le FPGA.

La sélection de la plate-forme dépend de facteurs tels que la performance, la consommation d'énergie et le coût par puce. La facilité des outils accompagnée d'une plate-forme spécifique pour aider les concepteurs à développer le système dans les limites du coût du système et du temps du projet joue également un rôle important.

L'objectif principal de l'utilisation des FPGAs pour l'implémentation des nœuds capteurs sans fil sur puce est de minimiser l'écart entre le développement de l'idée/algorithme et la conception du système embarqué. Aussi, l'FPGA présente d'autres avantages tels que, sa reconfigurabilité, son faible coût et son faible délai de mise sur le marché.

CHAPITRE 3 :

CONCEPTION ET IMPLEMENTATION DES NŒUDS D'UN RCSF SUR UNE PLATEFORME RECONFIGURABLE DE TYPE FPGA

3.1 Introduction

L'utilisation du circuit FPGA comme une plateforme reconfigurable peut améliorer les performances de traitement de nombreux systèmes, y compris les RCSFs. Dans ce cas, la conception d'un nœud peut bénéficier des technologies des plateformes reconfigurables en plusieurs aspects. L'implémentation sur un circuit FPGA offre une rapidité par rapport à celle basée sur un processeur. Non seulement la flexibilité inhérente à la conception est exploitée pour développer des accélérateurs matériels personnalisés, mais aussi à la possibilité de prototyper de nouvelles architectures de processeurs selon les exigences requises par l'application cible. Cependant, le coût de conception et d'implémentation reste une des restrictions des plateformes reconfigurables qui empêche son utilisation dans des domaines d'applications émanant du monde réel, limitant souvent les plateformes réalisées à des fins expérimentales.

L'objectif principal de ce travail est de développer un RCSF basé sur des circuits FPGAs et orienté vers une application industrielle. Le nœud capteur traite les données avant de les transmettre vers la station de base. Ce chapitre examine d'abord les travaux connexes afin de distinguer les projets avec les nœuds capteurs basés sur une plateforme reconfigurable exploités dans des applications réelles ou réalisés seulement au niveau prototype. Sur la base de cette analyse, notre conception et implémentation des nœuds est présentée. Les différents blocs constituant l'architecture proposée, ses principales caractéristiques, ainsi que les métriques de performance sont présentés et illustrés.

3.2 Travaux connexes

Cette section présente une synthèse des travaux de recherche récents liés aux possibilités d'intégrer les FPGAs dans les architectures des nœuds capteur sans fil ainsi que leurs applications. L'utilisation des FPGAs comme une solution dans les nœuds capteurs a débuté dès l'apparition des RCSFs avec des projets tel que le CA μ S [104] et l'architecture PicoRadio [105]. Malgré le succès apporté par les deux projets en termes de flexibilité et de faible consommation d'énergie, l'intégration des FPGAs n'a jamais été complètement exploitée. En 2003, Vieira et al. [106] ont souligné plusieurs limitations concernant cette solution. Tout d'abord, ils ont affirmé que la consommation d'énergie des FPGAs les rendait mal adaptés pour faire partie des nœuds capteurs. En outre, selon ces auteurs, les langages de description HDL tels que le Verilog et le VHDL ne peuvent pas correctement décrire la complexité d'un nœud capteur. Selon [107], l'architecture du circuit FPGA a changé depuis le début du 21^è siècle. Beaucoup d'améliorations ont été réalisées dans les FPGAs, et les principaux fabricants (Xilinx et Altera) se sont concentrés sur la réduction de la consommation d'énergie et l'optimisation du temps de développement. Aussi, grâce aux ressources embarquées incluses dans le FPGA telles que les DSP slices et les blocs mémoires rapides, elles sont actuellement utilisées pour l'accélération des algorithmes soit comme des coprocesseurs, soit en systèmes autonomes, c'est-à-dire System-On-Chips (SoCs).

Une étude approfondie sur la pertinence et les défis de l'utilisation des FPGAs dans plusieurs applications des RCSFs est traitée dans [108]. Il est souligné dans cette étude que les FPGAs sont particulièrement utiles pour les applications des RCSFs exigeantes.

Selon les auteurs de [108], les FPGAs attirent l'attention des développeurs des RCSFs grâce à leur performance et à leur capacité de reconfiguration par la reprogrammation. En plus, ils sont utiles surtout pour les applications complexes, où les implémentations sont utilisées pour un type spécifique d'applications des RCSFs en exploitant la puissance de calcul élevée des FPGAs.

Dans la littérature [108], [69] et [46], les FPGAs sont présents dans l'architecture des nœuds capteurs en trois formes différentes :

- Nœuds capteurs avec un circuit FPGA autonome,
- Nœuds capteurs basés sur une combinaison d'un microcontrôleur et d'un circuit FPGA comme coprocesseur,
- Nœuds capteurs à base d'un circuit FPGA pour le prototypage des systèmes sur puce.

Les sous-sections suivantes examineront ces différentes approches indépendamment.

3.2.1 Nœuds capteurs avec un circuit FPGA autonome

Dans cette architecture, les MCUs sont complètement remplacés par des circuits FPGAs. Cette solution influe considérablement sur l'efficacité énergétique du nœud et les FPGA peuvent épuiser rapidement la batterie, en plus elle réduit intrinsèquement la flexibilité de l'architecture. Le tableau 3.1 résume les travaux existants qui suivent cette approche. Ces travaux se divisent en deux catégories. La première concerne le prototypage des circuits ASICs [109][110]. Les résultats de ces projets montrent que la consommation est de 364.85 mW et 267 mW respectivement, ce qui est élevé si on les compare avec des nœuds commercialisés comme le Mica. Les projets de la deuxième catégorie sont eux orientés au déploiement réel avec des configurations spécifiques à l'application. On note ici que l'utilisation des FPGAs à consommation d'énergie ultra-basse (par exemple, Actel IGLOO) peut jouer un rôle important pour les applications à très faible consommation. C'est le cas des projets [111] et [112] où la consommation moyenne est de 8 mW et 4.2 mW pour les applications d'analyse des vibrations et le traitement d'image respectivement, ce qui présente une faible consommation énergétique par rapport aux projets de la première catégorie.

Tableau 3.1: Nœuds capteurs avec un FPGA autonome

Référence	Année	Plateforme	Module de Communication	Consommation d'énergie (mW)	Application
Ledeczi et al. [113]	2009	Actel Igloo	2.4 GHz	450 (150 mA / 3V)	Surveillance de l'état des structures
Garcia et al. [114]	2009	Xilinx Virtex-4 FX100	-	-	Suivi des cibles
Vera-salas et al. [111]	2010	Actel Igloo AGLN250ZV2	MRF24J40	8 (0.03 mode veille)	Surveillance industrielle
Kaddachi et al. [109]	2011	Xilinx Virtes-5 xc5vlx330t	-	364.85	Traitement d'image
Gasparini et al. [112]	2011	Actel Igloo AGL600	TI CC2420	4.2	Traitement d'image
Chefi et al. [110]	2011	Xilinx Virtes-5	-	267	Traitement d'image
Turcza et al. [115]	2011	Silicon Blue iCE65L08	-	12.5	Biomédicale
Pham et al. [116]	2011	Xilinx Spartan3 XC3S1000	Xbee	8.57	Traitement d'image
Philipp et al. [117]	2011	Xilinx	Personnaliser	-	Usage général
Philipp et al. [118]	2011	Actel Igloo AGL1000V5	TI CC2531	192	Localisation de la source acoustique
Liao et al. [119]	2013	Xilinx Spartan3 XC3S1500	TI CC2420	-	Les Tâches réseau
El kateeb et al. [120][121]	2013	Xilinx Spartan3 XC3S700A	-	-	Usage général
Nyländen et al. [68]	2014	Actel Igloo AGL1000	TI CC2420	40 – 70	Surveillance industriels

3.2.2 Nœuds capteurs basés sur une combinaison d'un microcontrôleur et d'un circuit FPGA comme coprocesseur

C'est la plus grande catégorie des nœuds à base d'un FPGA. Les MCUs qui sont présents couramment dans l'architecture des nœuds sont étendus avec des FPGAs afin de libérer le microcontrôleur des tâches spécifiques comme le traitement d'images [122] ou l'analyse des vibrations [123], qui ne peuvent pas être traitées efficacement par une implémentation software, en plus des restrictions de ressources dans les microcontrôleurs. L'utilisation de l'accélération matérielle par le circuit FPGA offre aux nœuds capteurs sans fil une meilleure efficacité que les MCUs autonomes, en particulier en termes de consommation d'énergie et de vitesse d'exécution. C'est la raison pour laquelle les unités de co-processing utilisées dans ces nœuds doivent être soigneusement choisies, afin de limiter la consommation d'énergie, et selon les exigences de l'application. Une liste des travaux basés sur cette approche est donnée dans le tableau 3.2.

Elle a été introduite au début des années 2000 par le projet PicoRadio [124]. L'objectif de ce projet est de développer un prototype de nœuds sans fil qui peut fournir toutes les fonctions de communication, de traitement et de géolocalisation nécessaires dans un réseau de capteurs. Un développement en trois phases des prototypes conduira au nœud final, le PicoNode III. Plus tard, de nombreux projets ont été créés avec des objectifs similaires. La majorité de ces projets sont basés sur une conception modulaire [72] [122], une tendance qui facilite l'utilisation des nœuds développés pour différentes applications. C'est le cas par exemple du projet Cookie mote [125][126], constitué d'une combinaison modulaire : une carte de traitement à base d'un FPGA, le CPU, la radio et le module de détection. En termes de consommation d'énergie, les plateformes FPGA à faible consommation comme l'Actel Igloo ou le Spartan-3 jouent un rôle important dans les projets récents. C'est le cas des projets [72][123][127][128], où la consommation moyenne est dans la gamme de 20 à 60 mW, une moyenne qui se situe dans les limites des nœuds commerciaux, ce qui stipule la possibilité de réaliser des nœuds sans fil plus efficaces via une combinaison d'un microcontrôleur et d'un circuit FPGA.

Tableau 3.2: Nœuds capteurs basés sur une combinaison d'un microcontrôleur et d'un circuit FPGA comme coprocesseur.

Référence	Année	FPGA	CPU	Module de communication	Consommation d'énergie (mW)	Application
Rabaey et al. [124]	2002	Xilinx XC4020XLA	Strong ARM SA-1110	Bluetooth	2600 (370 mA / 7V)	Les tâches réseau
Simon et al [129]	2004	Xilinx Spartan II	ATMega 128L	TI CC1000	-	Suivi des cibles
Bellis et al. [130] O'Flynn et al. [131]	2005	Xilinx Spartan II-E XC2S300E	ATMega128L	nRF2401	198 (66 mA / 3V)	Usage général
Caldas et al. [132]	2005	Xilinx Spartan II-E XC2S300E	TI MSP430F149	TI CC1000	-	Détection d'incendie
Virk et al. [133]	2005	Xilinx Spartan 3 XC3S400	ATMega 128L	nRF2401	-	Surveillance des animaux
Portilla et al. [125][126][134]	2007- 2012	Xilinx Spartan 3 XC3S200	ADuC831	Bluetooth	-	Usage général
Tanaka et al. [123]	2008	Actel Igloo AGL600	ATMega 644p	-	42.82	Traitement de signal
Koskinen et al. [135]	2010	Actel Igloo	AVR XMEGA	868 MHz	-	Surveillance industrielle
Berder et al. [72]	2010	Actel Igloo AGL125	TI MSP430	TI CC2420	25	Usage général
Sun et al. [122]	2011	Xilinx Spartan-3E XC3S100E	AT91SAM7X256	-	-	Multimédia
Krasteva et al. [127][128]	2011	Xilinx Spartan-3 XC3S200	ADUC841	ETRX2	60	Usage général
Hsieh et al. [136]	2012	Actel SmartFusion	ARM Cortex-M3	TI CC2520	72.9	Usage général
Shahzad et al. [137][138] Imran et al. [71]	2014	Xilinx XC6SLX16	Microchip AT32UC3B0512	TI CC2520	670	Usage général

3.2.3 Nœuds capteurs à base d'un circuit FPGA pour le prototypage des systèmes sur puce

Cette approche est basée considérablement sur la capacité du circuit FPGA ainsi que les outils de conception disponibles afin d'évaluer les différentes alternatives de conception. C'est une approche simple, économique et rapide. Le choix de la technologie FPGA a un impact direct sur les performances du nœud réalisé surtout sur la consommation énergétique, où les périphériques sont généralement sélectionnés avec un nombre élevé de ressources pour bénéficier de la flexibilité maximale lors de l'évaluation des architectures SoC. Des ressources suffisantes sur l'FPGA permettent l'implémentation d'une architecture SoC complète incluant un CPU, des mémoires et des composants périphériques.

Par conséquent, la consommation d'énergie augmente considérablement dans le nœud. Le choix d'un grand nombre de périphériques n'est pas toujours une solution idéale. En outre, il y a le risque que toutes les ressources disponibles ne soient pas exploitées, ce qui entraîne un gaspillage dans les ressources et par conséquent dans la consommation d'énergie. Cela explique l'absence des données relatives à la puissance consommée dans les travaux réalisés où la plupart ne considèrent pas les modes de fonctionnement à faible puissance en se concentrant uniquement sur les aspects de traitement. Seuls quelques travaux rapportent la consommation moyenne, et le meilleur résultat étant obtenu dans [139] (118 mW) où un SoC basé sur le hardcore PowerPC est utilisé pour réduire les opérations de traitement en implémentant un système dynamique d'agrégation de données. Les travaux les plus pertinents de cette approche, ainsi que les principales caractéristiques des nœuds proposés sont représentées dans le tableau 3.3.

En général, ces travaux se focalisent sur des modules périphériques personnalisés dans le SoC. Le CPU est généralement de type softcore disponible dans toutes les plateformes FPGA de Xilinx et Altera (MicroBlaze et Nios II respectivement). En plus, le soft-core est capable de communiquer avec des blocs personnalisés décrits en VHDL ou en Verilog, ce qui représente un avantage majeur dans la conception des nœuds.

Xilinx a récemment commencé à mettre sur le marché des plates-formes avec des processeurs hard. Par exemple, le modèle Zynq-7000 qui est basée sur un processeur dual Core Cortex A9, fournissant une plateforme puissante adapté aux systèmes sur puce. Malgré que son déploiement dans le développement des nœuds sans fil reste limité en à quelques projets, son utilisation représente une solution prometteuse pour les futures réalisations.

Les travaux représentés sur le tableau 3.3 montrent que les performances des nœuds en terme de traitement peuvent être largement améliorées en utilisant des accélérateurs matériels spécialisés comme c'est le cas pour les algorithmes de cryptage [140][141] et de traitement d'images [142]. Il est intéressant de noter que l'utilisation d'un FPGA pour faire un traitement simple n'est pas utile, encore plus si les FPGAs sont celles d'usage général.

Le travail réalisé dans cette thèse suit cette dernière approche et ce dans le but de proposer une plateforme adaptée à des applications industrielles, et bénéficiera des caractéristiques principales de conception des FPGAs tels que le partage de ressources et le prototypage d'architecture. Cette approche favorise aussi l'implémentation des processeurs en tant que soft-cores, ce qui permet une meilleure exploitation du circuit FPGA par l'usage d'une conception d'un système sur puce, malgré que cette solution reste une solution coûteuse en termes de consommation de ressources. Le reste de ce chapitre s'efforce à décrire la conception d'une architecture qui répond à ces objectifs.

3.3 Description des nœuds sans fil réalisés

Les exigences de conception et les fonctionnalités d'un nœud capteur sans fil basée sur le circuit FPGA ont été identifiées dans les chapitres précédents. Dans cette partie, l'architecture de base du nœud développé sera décrite, ainsi que les métriques d'évaluations de l'implémentation matérielle dans le cadre de ce projet de recherche.

Tableau 3.3: Nœuds capteurs à base d'un circuit FPGA pour le prototypage des systèmes sur puce

Référence	Année	FPGA	CPU	Module de communication	Consommation d'énergie (mW)	Application
Plessl et al. [143]	2003	Xilinx Virtex-II Pro	LEON	Bluetooth	-	Informatique vestimentaire
Volgyesi et al. [144]	2007	Xilinx Spartan3	PicoBlaze	Bluetooth	-	Classification des cibles
Ahola et al. [145]	2007	Altera Cyclone II	NIOS II	Bluetooth	330 (100 mA)	Informatique vestimentaire
Hinkelmann et al. [146][147][148]	2008	Xilinx Spartan3	LEON	XEMICS DP-1203	700 – 1100	Usage général
Commuri et al. [139]	2008	Xilinx Virtex-II Pro	PowerPC	Wi232 DTS	118.01	Agrégation de données
Chalivendra et al. [149]	2008	Altera Cyclone	NIOS II	Bluetooth	-	Usage général
Muralidhar et al. [150]	2008	Altera	NIOS II	Bluetooth	-	Usage général
Hu et al. [142]	2009	Altera Cyclone II	NIOS II	nRF24L01	-	Multimédia
Lu et al. [151]	2009	-	MC8051	-	450	Surveillance du trafic
Allgayer et al. [152]	2009	FPGA Virtex-II	-	TI CC2420	-	Prototypage
Tong et al. [140]	2009	Xilinx Spartan-3 ^E	Microblaze	TI CC2430	-	Cryptage
Wei et al. [153]	2009	Xilinx Cyclone II	OpenRISC	-	221	Usage général

Liu et al. [154]	2010	Xilinx Spartan-3 ^E	Microblaze	-	2850	Surveillance des animaux
Li et al. [141]	2012	Xilinx Virtex 4	LEION3 / 8051	-	353	Cryptage
Hayek et al.	2012	Xilinx Spartan3	8051	OWSPA311g	-	Usage général
Sumarudin et al. [155]	2014	Xilinx Zynq-7000	Cortex-A9	WiFi	-	Usage général
Gaga et al. [156]	2015	Xilinx Spartan-3 ^E	Microblaze	Xbee Pro	-	Usage général
Zhai et al. [157]	2015	Xilinx Zynq-7000	Cortex-A9	-	1143	Collecte de données environnementales
Keerthana et al [158]	2016	Xilinx Zynq-7000	Cortex-A9	-	-	Usage général
Zhou et al. [159]	2016	Xilinx Zynq-7000	Cortex-A9	Xbee	-	surveillance de l'environnement
Bengherbia et al. [160]	2016	Xilinx Artix-7	Microblaze	Xbee Pro	-	Usage général

Sur la base des considérations précédentes, nous avons opté pour la plateforme Basys-3 (Figure 3.1) de chez Digilent. Cette carte est bâtie autour du circuit FPGA de Xilinx de type Artix-7 XC7A35T-1CPG236C, qui possède des ressources internes et des capacités appréciables, avec un faible coût. Les ressources disponibles sont présentées dans le tableau 3.4.

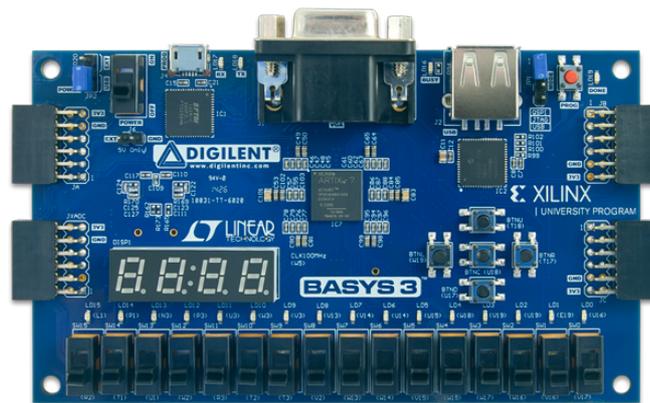


Figure 3.1: La plateforme Digilent Basys-3

La plateforme comprend aussi des commutateurs, des diodes électroluminescentes et d'autres périphériques d'E/S. En plus, des connecteurs auxiliaires qui ajoutent à la carte des possibilités d'extensions avec des modules additionnels sans avoir besoin de matériel supplémentaire.

Tableau 3.4: Ressources du circuit FPGA utilisé dans la plateforme BASYS-3

Basys – 3	Slices	Logic Cells	Bloc RAM	DSP Slices	Prix
XC7A35T-1CPG236C	5200	33280	1800 Kbit	90	150 \$

Le Basys3 fonctionne avec le nouveau logiciel de développement : Vivado™ Design Suite de Xilinx. Il comprend plusieurs nouveaux outils qui facilitent et améliorent les dernières méthodes de conception. Il fonctionne plus rapidement,

permet une meilleure utilisation des ressources FPGA et permet aux concepteurs de concentrer leur temps à évaluer différents choix de conception.

3.3.1 Architecture globale du nœud

L'architecture générale d'un nœud capteur sans fil reste inchangée, mais ses caractéristiques varient largement en fonction de l'application. La figure 3.2 montre l'architecture des nœuds capteurs sans fil proposés. La conception doit atteindre un certain nombre d'objectifs : une architecture reconfigurable à faible puissance avec une transmission sans fil efficace, une convenance pour une surveillance des machines industrielles en termes de fréquence d'échantillonnage, de traitement des données et de synchronisation.

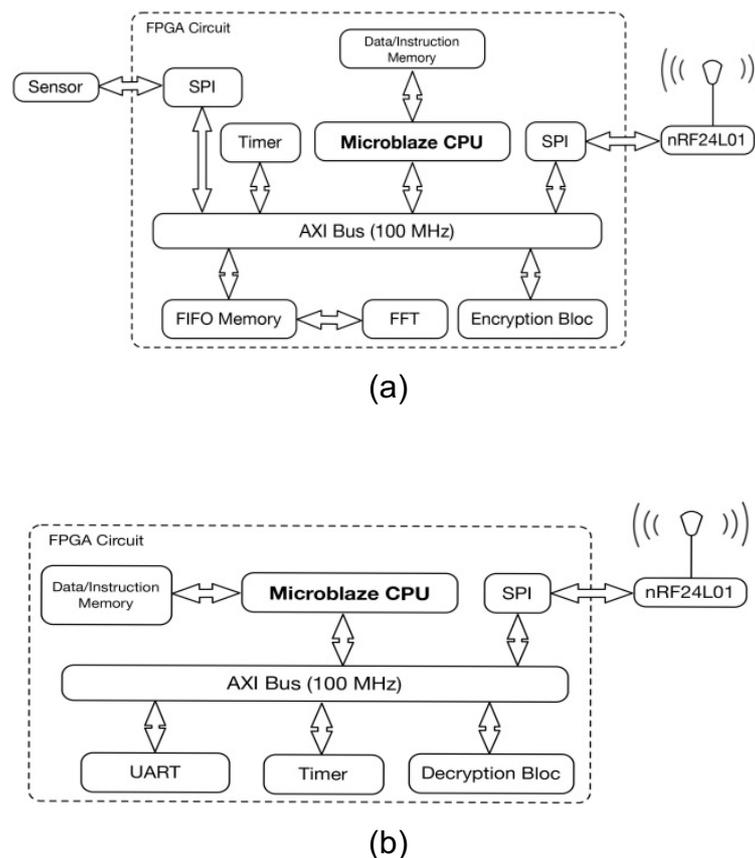


Figure 3.2: Architecture des nœuds sans fil : (a) nœud capteur; (b) nœud Sink

Dans la figure 3.2, les blocs implémentés sur le circuit FPGA de la plate-forme Basys-3 sont limités par des lignes en pointillés. Ces blocs sont ajoutés sous forme

de IP-cores personnalisés dans la conception. Ils sont interconnectés par le biais du bus AXI.

En résumé, l'architecture se compose d'un processeur soft-core (MicroBlaze), de la mémoire, des IPs personnalisées pour le traitement FFT et la sécurité, ainsi que des interfaces de type « SPI » reliant les modules de communication et le module d'acquisition pour le nœud capteur.

La spécification AXI décrit des protocoles de transfert de données entre les différents IPs à l'aide d'une norme de signalisation définie. Il assure également un moyen efficace et flexible pour le transfert de données.

3.3.2 L'unité de contrôle

Considérée comme le noyau du nœud capteur sans fil, ce module permet de gérer les différentes tâches du nœud telles que : l'acquisition de données, la gestion des traitements et des protocoles de communication, le transfert de données et la gestion de la mémoire.

Nous avons opté pour un processeur soft-core de type MicroBlaze fonctionnant à 100 MHz. Xilinx a proposé le MicroBlaze particulièrement pour la conception des SoCs, dans le but d'offrir aux concepteurs la flexibilité de sélectionner toute combinaison de périphériques afin de réduire le taux d'occupation dans le circuit FPGA et le coût de la conception en faisant des compromis entre les solutions matérielles et logicielles.

Le processeur MicroBlaze est un processeur RISC à 3 étages avec une architecture Harvard et des registres internes de 32 bits. Il dispose d'un bus d'instructions et de données internes et externes séparées (ILMB, DLMB, IOPB et DOPB). Il est entièrement intégré dans l'environnement de conception des FPGAs de Xilinx. Il peut être facilement configuré pour de nombreuses applications différentes. Il possède plus de 70 options de configuration, permettant à l'utilisateur de sélectionner ou de modifier les composants internes selon les besoins de l'application. Dans notre cas, de nombreux périphériques sont ajoutés avec le

MicroBlaze, afin de constituer un microcontrôleur complet et personnalisable. Il y a, entre autres : un contrôleur mémoire (SRAM, Flash), un contrôleur mémoire SDRAM, un UART lite, un Timer avec une fonction PWM, une interface SPI, un contrôleur d'interruptions, et un GPIO (interface d'entrées-sorties génériques).

La figure 3.3 représente le diagramme en blocs du MicroBlaze. Les parties en couleur grise sont optionnelles et montrent à quel point le MicroBlaze est configurable. Les caches permettent au processeur un gain en temps d'exécution en récupérant des données directement à l'intérieur sans aller les chercher dans la mémoire. Le « Barrel Shift », le « Multiplieur », le « Diviseur » et le « FPU : Floating Point Unit » permettent d'accélérer les traitements de données.

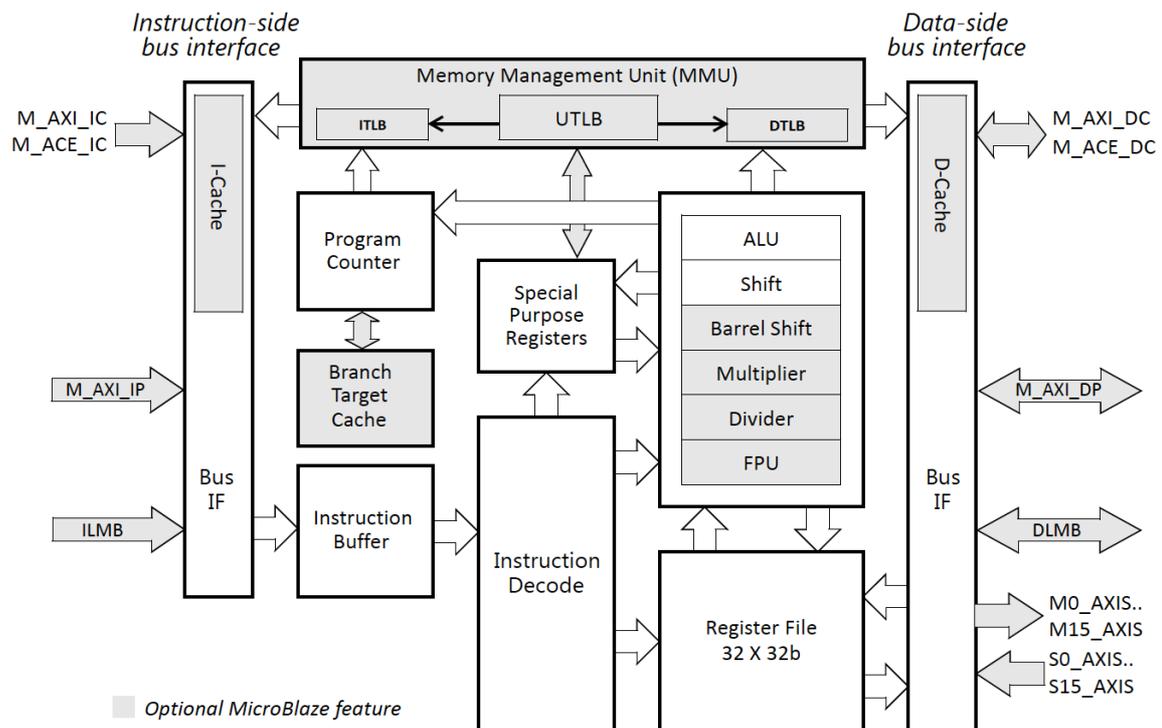


Figure 3.3: Diagramme en bloc du MicroBlaze

Ce diagramme met aussi en évidence les 5 étages du « pipeline » :

1. La lecture de l'instruction dans le buffer d'instruction.
2. Le décodage de l'instruction dans le décodeur d'instruction

3. L'accès mémoire pour la lecture des opérandes
4. Le calcul ou l'exécution de l'instruction dans l'ALU
5. L'écriture du résultat dans la mémoire spéciale.

3.3.3 Bus et protocole AXI

Le MicroBlaze peut être connecté à n'importe quel composant dans la zone logique programmable à travers l'ensemble des bus AXI (Advanced eXtensible Interface). Le protocole AXI, est une norme pour la connexion, la gestion et la communication des blocs dans les modèles SoC. L'objectif principal est de faciliter la gestion et le développement des systèmes multi-processeurs, de ses contrôleurs et ses périphériques. Ce protocole est très robuste et garantit une communication sûre et rapide entre les différents modules liés à un système d'interface maître-esclave. Du point de vue pratique, le protocole AXI est utilisé par la plupart des IPs de Xilinx. Chaque module basé sur l'AXI possède son propre pilote, qui peut être initialisé dans le code source en exécutant des fonctions spécifiques. Par conséquent, ce protocole est fondamental pour la configuration des paramètres matériels, la gestion interne des données et le contrôle complet du matériel. Il existe trois types d'interfaces AXI :

- AXI-Full : Pour les exigences des mémoires mappées de haute performance
- AXI-Lite : Pour les simples accès mémoire, à faible débit.
- AXI-Stream : pour des flux de données rapides en continu

3.3.4 L'unité de traitement FFT

La FFT est un algorithme numérique efficace pour le calcul de la transformée de Fourier discrète (DFT) dont le nombre d'échantillons doit être une puissance entière positive de 2.

Soit la séquence de N échantillons $x(n)$, pour calculer une TFD, on doit calculer N valeurs $X(k)$:

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad 0 \leq k \leq N-1 \quad (3.1)$$

Avec

$$W_N^{nk} = \exp\left(-j2\pi \frac{nk}{N}\right) \quad (3.2)$$

Où $x(n)$ est le signal d'entrée à temps discret et $X(k)$ sa DFT, n représente l'indice du domaine temporel discret, alors que k est l'indice du domaine fréquentiel normalisé.

Parmi les approches utilisées pour le calcul de la FFT, on peut citer par exemple l'algorithme de Cooley-Tukey, l'algorithme Good-Thomas et l'algorithme Winograd. L'algorithme Cooley-Tukey est le plus populaire, il a été largement utilisé pour de nombreuses applications pratiques en raison de ses avantages d'implémentation. L'algorithme décompose le module DFT en petits blocs de calcul, appelés radix-2, qui consistent en un traitement simple d'additions et de multiplications, appelés papillons « butterflies ». Il existe deux formes d'algorithmes pour le radix-2, l'algorithme de décimation en temps (DIT) et la décimation en fréquence (DIF). Ces deux algorithmes ont la même complexité mais diffèrent dans l'arrangement du calcul (Figure 3.4).

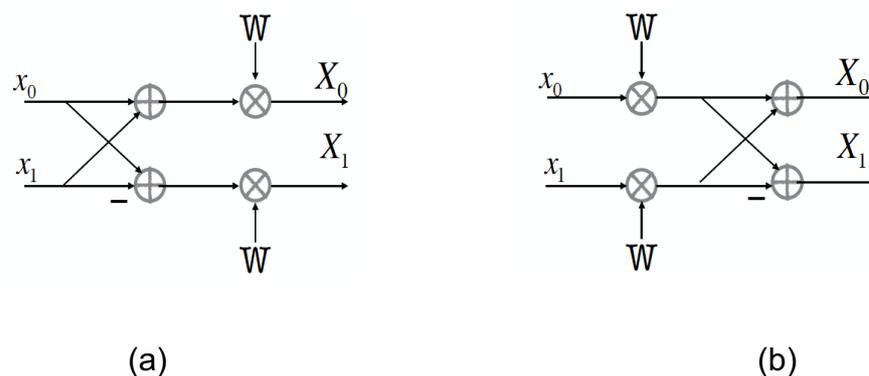


Figure 3.4: Le bloc radix-2 (a) radix-2 DIF butterfly (b) radix-2 DIT butterfly

Dans les applications des RCSFs, la FFT représente un algorithme de référence pour comparer les performances des nœuds en puissance de traitement. Les

noeuds ne sont pas adaptés généralement aux algorithmes FFT en raison des limitations en calcul intensif dans les opérations de traitement numérique des signaux, ainsi qu'en mémoire. Dans le cas de l'analyse des vibrations, l'analyse par la FFT facilite la détection des pannes sur l'espace des fréquences pour les signaux stationnaires et stables. Un tel algorithme est également couramment utilisé dans les applications de traitement du signal audio ou les systèmes biomédicaux intelligents.

Dans la présente conception, nous avons opté pour l'IP-core « FFT » de Xilinx. Cet IP-core offre quatre architectures pour avoir la possibilité de faire un compromis entre la taille du core et le temps de traitement FFT.

- Pipelined Streaming I/O : Permet le traitement continu des données.
- Radix-4 Burst I/O : Charge et traite les données séparément, en utilisant une approche itérative. Il est plus petit en taille que la solution « pipeline », mais a un temps de traitement plus long.
- Radix-2 Burst I/O : Utilise la même approche itérative que Radix-4, mais l'étage « papillon » est plus petit. Cela signifie qu'il est plus petit que la solution Radix-4, mais le temps de traitement est plus long.
- Radix-2 Lite Burst I/O : Basé sur l'architecture Radix-2, cette variante utilise une approche de multiplexage temporelle dans les étages « papillon » pour un noyau encore plus petit, au prix d'un temps de traitement plus long.

Compte tenu des contraintes de ressources du circuit FPGA choisi, nous avons utilisé une version FFT de 1024 points avec une architecture en « pipeline streaming » qui convient au traitement en temps réel. Le rôle de ce bloc est de traiter les 1024 échantillons (32 bits/échantillon), provenant de l'unité d'acquisition. Les données sont stockées dans une mémoire FIFO avant d'être traitées par le bloc FFT. La même mémoire est utilisée pour stocker le résultat de sortie de l'unité FFT après traitement.

Cet IP-core a la possibilité d'effectuer simultanément des calculs de la FFT sur la trame de données courante, de charger les données d'entrée pour la trame suivante, et décharger les résultats de la trame précédente.

Il faut noter ici l'utilisation de l'IP-core « AXI streaming FIFO » de Xilinx, comme une solution pour que le bloc FFT travaille en mode streaming. L'IP-core « AXI streaming FIFO » permet d'écrire ou de lire des paquets de données vers ou à partir d'un périphérique sans aucune préoccupation par rapport à l'interface « AXI Streaming ».

La solution « Pipelined Streaming » regroupe plusieurs cellules de traitement en papillons Radix-2 offrant un traitement continu des données. Chaque cellule dispose de ses propres blocs de mémoire pour stocker les données d'entrée et intermédiaires (Figure 3.5).

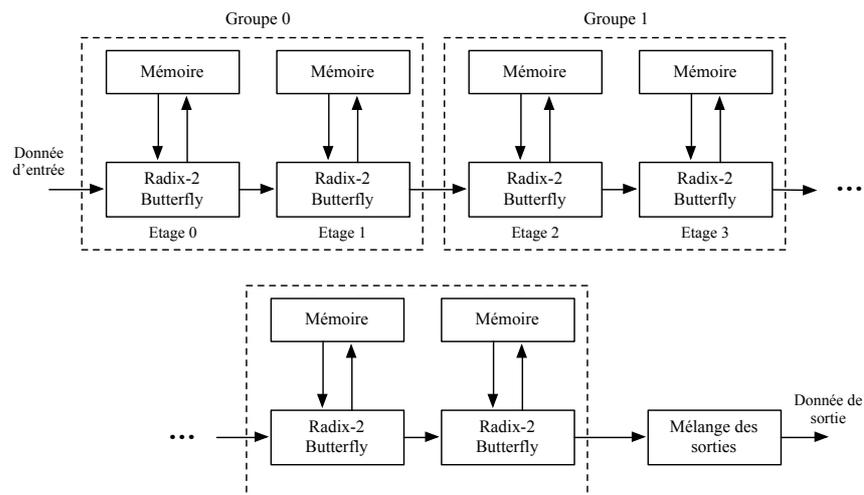


Figure 3.5: Architecture “Pipelined Streaming I/O” du bloc FFT

3.3.5 L'unité de sécurité

En 2001, l'algorithme de Rijndael a été sélectionné par l'institut national des standards et de la technologie (NIST) en tant qu'une nouvelle norme avancée de cryptage « AES » [161]. Auparavant, de nombreuses implémentations sur des FPGAs de l'AES étaient proposées et évaluées. Jusqu'à présent, la plupart de ces implémentations sont adaptées uniquement aux applications haut de gamme et non

aux applications des RCSFs en raison des coûts élevés. Selon [162], l'échange sécurisé de données entre les différents nœuds du RCSF deviendra de plus en plus important, et la nécessité d'étendre l'AES à ce type de dispositif devient primordiale. Cependant, l'implémentation par un microcontrôleur de tels algorithmes présente certaines limitations, le microcontrôleur pouvant rapidement être surchargé et sa batterie peut être rapidement épuisée, surtout dans le cas d'une application ou d'un échange considérable de données cryptées/décryptées doit être effectué [163]. Pour remédier à ces limites, l'émergence du circuit FPGA et l'introduction de ressources spécifiques telles que les « DSP slices » donnent lieu à un certain nombre de recherches dans ce domaine, comme dans le cas de l'AES dans [164] et l'implémentation de l'algorithme de cryptographie ECC dans [165], où ces ressources peuvent être utilisées pour accélérer les opérations arithmétiques.

Basé sur l'algorithme de Rijndael pour une clé de 128 bits, nous avons conçu et implémenté deux IP-cores personnalisés assurant les modules de cryptage et de décryptage AES en mode pipeline. Chacun de ces deux modules est créé séparément et ce avec ses propres sous-modules dans le but de faciliter l'intégration de l'un de ces deux blocs en fonction des besoins, de libérer plus de ressources dans le circuit FPGA, d'accélérer le processus de cryptage/décryptage, et de donner plus de flexibilité au nœud capteur. Les différents blocs constituant ces deux IP-cores sont décrits en langage VHDL, ils incluent une interface AXI afin d'être interconnectés avec le processeur MicroBlaze.

3.3.5.1 Conception des modules de l'algorithme AES

L'organigramme fonctionnel de la fonction de cryptage/décryptage est illustré sur la figure 3.6. L'algorithme prend comme entrée un message de 128 bits (16 octets), avec une clé de 128 bits. Pour cette clé, l'AES nécessite 10 tours pour accomplir l'opération de cryptage/décryptage. Les 16 octets d'entrée sont placés dans une matrice de 4x4 éléments et sont permutés par la suite selon une table prédéfinie. A chaque rond, quatre transformations sont appliquées :

- 1) Substitution d'octets dans le bloc « SubBytes » ;

- 2) Décalage de rangées dans le bloc « ShiftRows » ;
- 3) Déplacement de colonnes (sauf à la dernière ronde) dans le bloc « MixColumns » ;
- 4) Addition d'une "clef de tour" qui varie à chaque tour « AddRoundKey ».

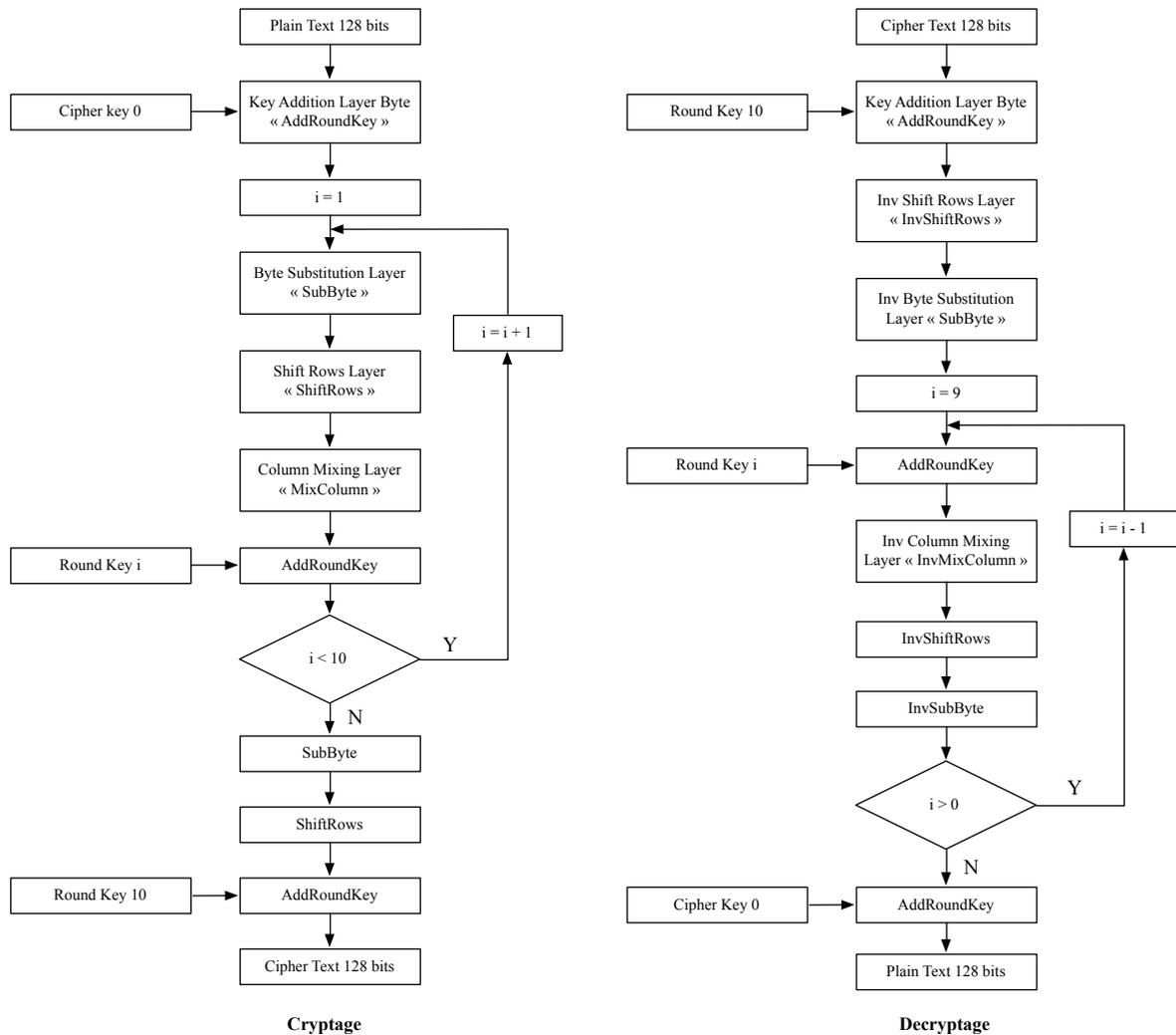


Figure 3.6: L'organigramme de l'algorithme AES

Le module de cryptage se compose des différents sous-modules suivants (Figure 3.7):

- Sous-module KeyGen : ce module fait l'extension de la clé. Il reçoit comme entrée la clé initiale Key0, puis génère les clés des 10 tours. Chaque clé générée est obtenue en passant par les étapes suivantes :

- a. Diviser la clé de 128 bits en quatre (04) mots de 32 bits k_0, k_1, k_2 et k_3 .
- b. Le premier mot de la clé à générer k'_0 est obtenu en passant par les opérations suivantes :
 - Un décalage à droite des octets du mot de 32 bits k_3 .
 - Le résultat passe par la table S-Box pour faire la substitution des quatre octets en parallèle.
 - Le premier octet à gauche de la sortie de substitution fait un XOR avec la valeur de la case correspondante du vecteur R_{con} . La table du vecteur de R_{con} est la suivante (Tableau 3.5) :

Tableau 3.5: Table R_{con} utilisée dans l'extension des clés

r	1	2	3	4	5	6	7	8	9	10
$R_{con}(r)$	0x01	0x02	0x04	0x08	0x10	0x20	0x40	0x80	0x1b	0x36

La règle de construction de cette table est :

$$\begin{cases} R_{con}(1) = 1 \\ R_{con}(r) = 2 * R_{con}(r - 1) \quad \text{pour } r > 1 \end{cases} \quad (3.3)$$

- c. Les trois derniers mots de la clé à générer (k'_1, k'_2 et k'_3) sont obtenus à partir d'un XOR des deux mots : le mot de la clé précédente k_i avec le précédent mot de la clé k'_{i-1} ($i=1, 2$ ou 3).

Ces clés sont stockées dans une RAM de 160 mots de 8 bits. Lorsque toutes les clés sont générées, le bit KeyVal passe à l'état '1', de sorte que le contrôleur pourra utiliser les nouvelles clés pour chaque tour.

- *Sous-module AddRoundKey* : Dans le module AddRoundKey, une opération XOR est effectuée entre la clé et les données, la clé étant sélectionnée selon le

tour. Ce module comporte deux entrées de 128 bits pour les données et la clé, un bit *AddVal* généré par le contrôleur qui indique la présence d'une nouvelle donnée. Le résultat de l'opération est donné sur 128 bits.

- *Sous-module SBox* : ce module effectue une substitution de données d'entrée 32 bits en utilisant une table spécifique de 256 mots de 8 bits (la table *Sbox*). La table est divisée en 16 lignes et 16 colonnes. Les données d'entrée sont divisées en quatre mots de 8 bits où chaque mot est utilisé pour sélectionner une cellule dans la table *Sbox*, dont les 4 premiers bits les moins significatifs représentent l'indice de la colonne dans la table *Sbox*, alors que les 4 bits les plus significatifs représentent l'indice de la ligne. La cellule sélectionnée est la nouvelle donnée de 32 bits qui est fournie à la sortie de ce module.
- *Sous-module ShiftRow* : dans ce module, les octets des trois dernières lignes de la table (message) sont cycliquement décalés à la gauche.
- *Sous-module MixColumns* : Ce module traite un mot d'entrée de 32 bits qui multiplie ses octets dans le champ fini de Galois $GF(2^8)$ par une matrice prédéfinie. La sortie obtenue est de 32 bits.
- *Le contrôleur* : Assure la synchronisation entre les différents sous-modules de cryptage en mode pipeline afin de réduire le temps de cryptage : *AddRoundKey*, *SBox* et *MixColumns*. De plus, il sélectionne le tour à exécuter, le mot à substituer, le mot à enregistrer pendant un tour, l'indication de démarrage de *AddRoundKey*, l'activation de *MixColumns*, ainsi que la validation de la sortie.

Le principe de fonctionnement du module de cryptage est le suivant : pour chaque tour, une clé spécifique est générée par le sous-module *KeyGen* afin de faire l'opération *AddRoundKey* entre les données et la clé. Les 128 bits résultants de cette dernière opération seront transférés au sous-module *ShiftRow* pour faire une opération de décalage. Le résultat du sous-module *ShiftRow* est divisé en quatre mots de 32 bits, chaque mot sera attribué au sous-module *SBox* pour la substitution et le résultat sera attribué au sous-module *MixColumns*. Les quatre mots de 32 bits du message seront stockés dans un registre de 128 bits (*Reg32128bits*) pour le tour suivant. Ces opérations sont réalisées de la même manière pour les 9 premiers tours. Au 10^{ème} tour, l'opération *MixColumns* n'est pas réalisée. Après le 10^{ème} tour,

le résultat final (DATAOUT) est obtenu avec une ligne de validation Outvalid sur le sous-module de AddRoundKey de sortie.

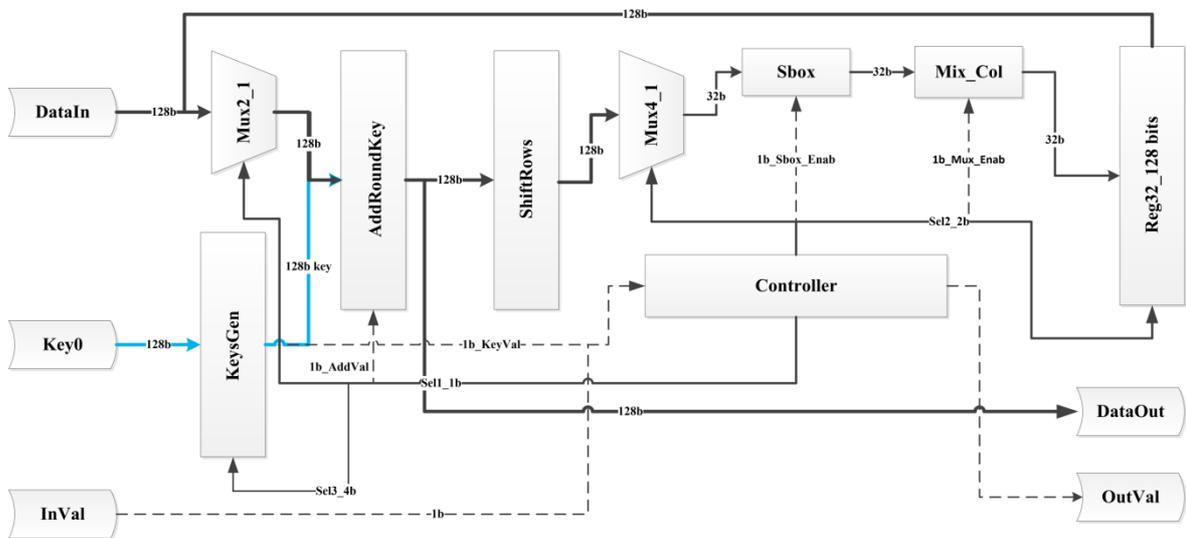


Figure 3.7: Diagramme en blocs du module de cryptage réalisé

Le module de décryptage suit les mêmes étapes de cryptage sauf que tous les sous-modules sont inversés (Figure 3.8). Le principe de fonctionnement du circuit est comme suite : A chaque tour, il faut une clé de tour générée par le sous-module KeyGen utilisée par l'opération AddRoundKey entre la donnée et cette clé. Le résultat de ce dernier se divise en quatre mots de 32 bits, chaque mot va s'affecter au sous-module InvMixColumns puis le résultat obtenu passe au sous-module InvSBox. Ces derniers sont mémorisés dans un registre de 128 bits (Reg32128bits) pour faire l'opération InvShiftRow. Le résultat de cette opération est envoyé au tour suivant. Pour le 1^{er} tour, l'opération InvMixColumns n'est pas faite. Enfin, le résultat final (Dataout) est obtenu avec une validation Outvalid sur la sortie du sous-module AddRoundKey.

3.3.5.2 Simulation des modules de cryptage/décryptage

L'architecture matérielle proposée pour les deux modules d'AES est décrite en langage VHDL. Pour chaque module, un test a été effectué pour vérifier son fonctionnement et calculer le nombre de cycles d'horloge nécessaires.

Les modules de cryptage et de décryptage sont simulés à l'aide de Modelsim 10.1c en envoyant 128 bits de données et une clé de 128 bits, suivis d'un bit '1' dans les entrées KeyValid et DataInvalid qui sont des entrées implémentées dans les deux modules pour indiquer la présence d'une nouvelle donnée ainsi que la clé. Une fois que le processus de cryptage est entamé, le bit OutValid est mis à '0'. Une fois que le processus de cryptage est terminé, le bit OutValid est mis à '1'.

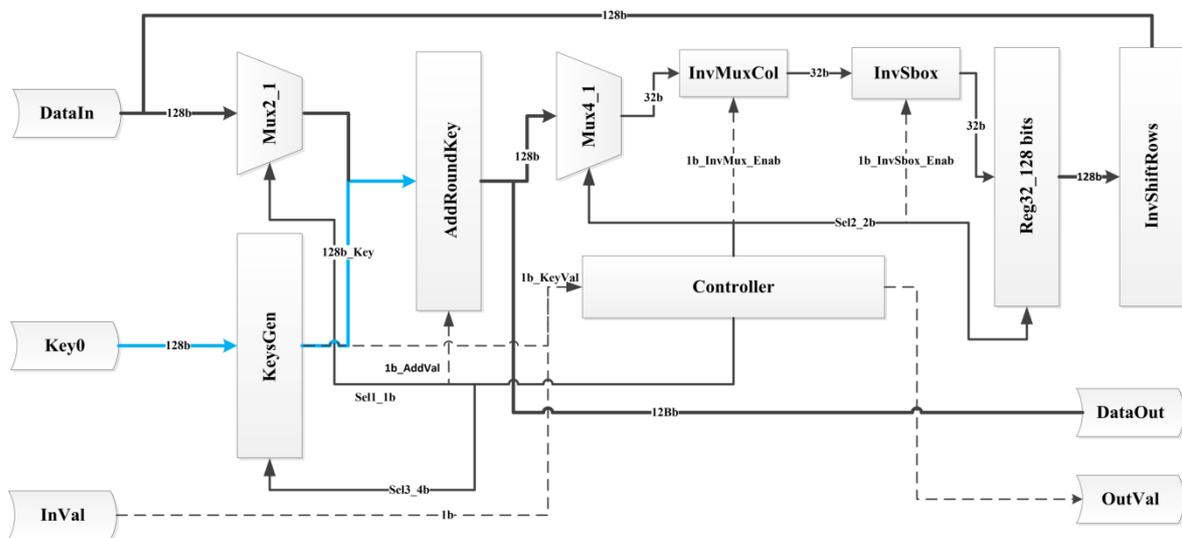


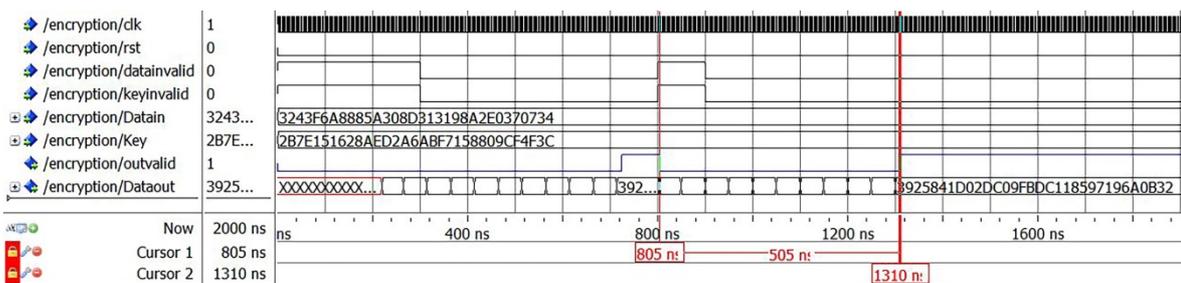
Figure 3.8: Diagramme en blocs du module de décryptage réalisé

Le bit OutValid a deux rôles. D'une part, il indique la présence de données chiffrées, d'autre part, il permet de mesurer le temps requis pour le cryptage en mesurant le temps entre le front descendant et le front montant. Le même processus de simulation est utilisé pour le décryptage. Les résultats de la simulation (Figure 3.9) montrent un temps de cryptage/ décryptage d'environ 505 ns pour une fréquence d'horloge de 200 MHz, ce qui équivaut à 101 cycles, donc un débit de cryptage de 29,09 Mo/s. Ceci est très suffisant pour les applications des RCSF.

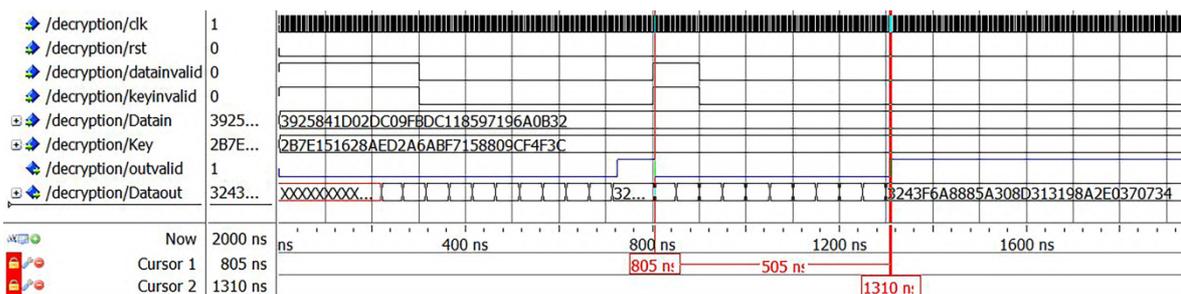
3.3.5.3 Synthèse et implémentation des IP-cores de cryptage/décryptage

Après avoir testé les deux modules de cryptage/décryptage avec le logiciel ModelSim, nous avons axé notre effort au développement de leur IP-cores, afin de

les intégrer après avec les autres unités du nœud capteur. Nous avons choisi une architecture à base des registres de 32 bits, une taille qui est fixée pour toutes les interconnexions AXI. Pour cela, nous avons utilisé 4 registres pour les données d'entrée (message à crypter dans le cas de cryptage ou le message crypté dans le cas de décryptage), 4 registres pour les données de sortie (message crypté dans le cas de cryptage ou le message clair dans le cas de décryptage) et 4 registres pour la clé de cryptage/décryptage. Des registres additionnels sont utilisés pour les différents bits de contrôle (DataValid, KeyValid, et OutValid). Les deux modules de cryptage et de décryptage sont synthétisés et ajoutés au SoC à base du MicroBlaze en tant que périphériques AXI. Le rapport détaillé des résultats concernant l'occupation de surface pour les différents blocs de l'architecture est reporté dans le tableau 3.6.



(a)



(b)

Figure 3.9: Résultats de simulations AES: (a) Cryptage (b) Décryptage

Le cryptage fonctionne comme suit. Tout d'abord, le MicroBlaze envoie par le biais du bus AXI un texte en claire de 128 bits vers l'IP-core de cryptage, ces données sont stockées dans 4 registres de 32 bits (DataIn), et la même chose pour la clé de

cryptage. Une fois que le texte en clair et la clé sont présents à l'entrée, le MicroBlaze met à '1' les bits de contrôle (DataInValid et KeyValid) afin que le bloc de cryptage commence le traitement. À la fin du traitement, les données cryptées sont stockées sur les 4 registres de sortie et le signal OutValid change son état de '0' à '1' indiquant la fin du traitement et la présence des données traitées. Le décryptage fonctionne de manière similaire.

Tableau 3.6: Consommation de ressources des IP-cores de cryptage/décryptage

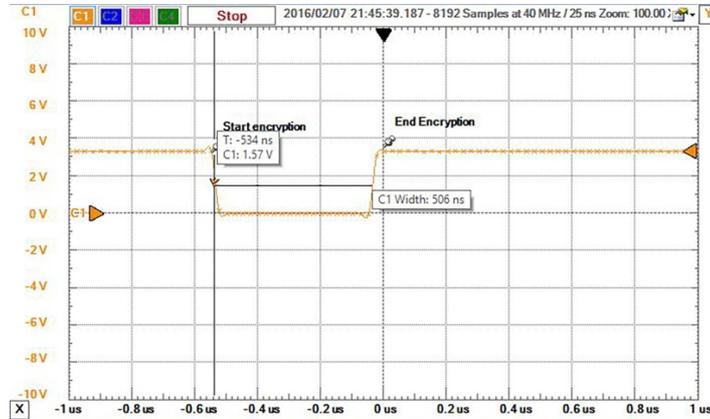
Ressources	Disponible	Cryptage		Décryptage	
		Utilisé	Utilisation (%)	Utilisé	Utilisation (%)
Slice LUT	20800	982	4.72	1061	5.10
Bascules	41600	1094	2.63	1100	2.64
Bloc RAM	50	1	2	1	2
Fréquence	200 MHz				

Pour mesurer le temps de traitement, nous avons visualisé sur l'oscilloscope le signal OutValid. Les résultats pratiques sont représentés sur la figure 3.10. Comme on peut le voir, le temps de cryptage est de 506 ns et le temps de décryptage est de 508 ns, valeurs qui sont très proches des résultats de la simulation. Il est à rappeler que nous avons également utilisé une fréquence de fonctionnement maximale de 200 MHz avec la carte Digilent Basys3.

3.3.6 Module de communication sans fil

Considérée comme l'unité la plus consommatrice d'énergie, le choix et la sélection des modules radio doivent répondre à certaines tendances technologiques telles que leur sensibilité au bruit et à l'interférence, la taille, le coût et leur consommation d'énergie qui doit être aussi faible que possible. L'émetteur-récepteur radio nRF24L01+ de chez NORDIC Semiconductor Corporation remplit la majorité des considérations précédentes (Tableau 3.7). Il est conçu pour assurer des communications à très grande vitesse (jusqu'à 2 Mbit/s) avec une puissance extrêmement faible (le courant RX ne dépasse pas 12,5 mA).

L'émetteur-récepteur se compose d'un synthétiseur de fréquence entièrement intégré, d'un amplificateur de puissance, d'un oscillateur à quartz, d'un modulateur/démodulateur, et d'un accélérateur protocolaire avancé « ShockBurst ».



(a)



(b)

Figure 3.10: Temps de traitement : (a) opération de cryptage (b) opération de décryptage

Le protocole « ShockBurst » permet de décharger le processeur des opérations de bas niveau pendant les temps critiques, ouvrant la voie à l'implémentation d'une connectivité sans fil avancée et robuste. La puissance de sortie, les canaux de fréquence et la configuration du protocole sont facilement programmables grâce à un bus SPI. Les modes intégrés tels que Power Down (courant 400 nA) et Standby

(32 μA à 130 μs de réveil) rendent la tâche d'économie d'énergie plus significative et facilement réalisable.

Dans ce travail, nous avons choisi le nRF24L01+ comme émetteur-récepteur RF. Le réseau est configuré en topologie étoile centralisée, qui est la topologie la plus commune dans les applications industrielles des RCSFs [7][166]. Afin de satisfaire ces exigences, nous avons utilisé la solution « MultiCeiver » qui est une solution matérielle innovante proposée comme une fonctionnalité dans le chipset nRF24L01+.

Tableau 3.7: Comparaison des modules RF

	nRF24L01+	TI CC2430	Xbee pro
Consommation de courant (Tx, Rx)	(11,3 mA, 12,3 mA)	(27 mA, 27 mA)	(150 mA, 55 mA)
Consommation en mode veille	0,1 μA	0,5 μA	< 10 μA
Débit de transmission	250 kbps, 1 Mbps, 2 Mbps	250 kbps	250 kbps

Avec cette fonctionnalité, jusqu'à six nœuds capteurs configurés comme des émetteurs peuvent communiquer avec un nœud puits (collecteur) configuré en tant que récepteur dans un canal de fréquence. Le nœud récepteur peut distinguer les nœuds capteurs en fonction de leurs différentes adresses physiques. En outre, nous avons configuré le nRF24L01+ pour gérer un « payload » dynamique de 32 octets et un débit de 2 Mbps. Tous ces paramètres ont été fixés dans le registre de configuration. La figure 3.11 représente un organigramme simplifié pour les deux modes de communication (Tx et Rx) implémentés dans la présente réalisation. En mode émetteur, le nRF24L01+ attend en mode veille jusqu'à ce qu'un message provienne de la couche d'application indiquant la présence des données à transmettre. Le message est transmis après la fin de la phase de configuration en mode Tx. Si la génération des paquets ACK (Acknowledgement) est activée, l'émetteur passe en mode récepteur Rx en attente de recevoir l'ACK du récepteur

et déclenche un compteur de temporisation. Lorsqu'un paquet ACK est reçu dans la période de réception des ACK, le module radio passe en mode veille. Si aucun ACK n'est reçu, la transmission est redémarrée jusqu'à ce que le nombre maximal de tentatives de retransmission soit atteint. En mode récepteur, le nRF24L01+ reste dans cet état jusqu'à ce qu'un message soit reçu de la couche physique. Si la génération des paquets ACK est activée, le mode Tx est activé pour la transmission du paquet ACK, sinon il reste en mode Rx.

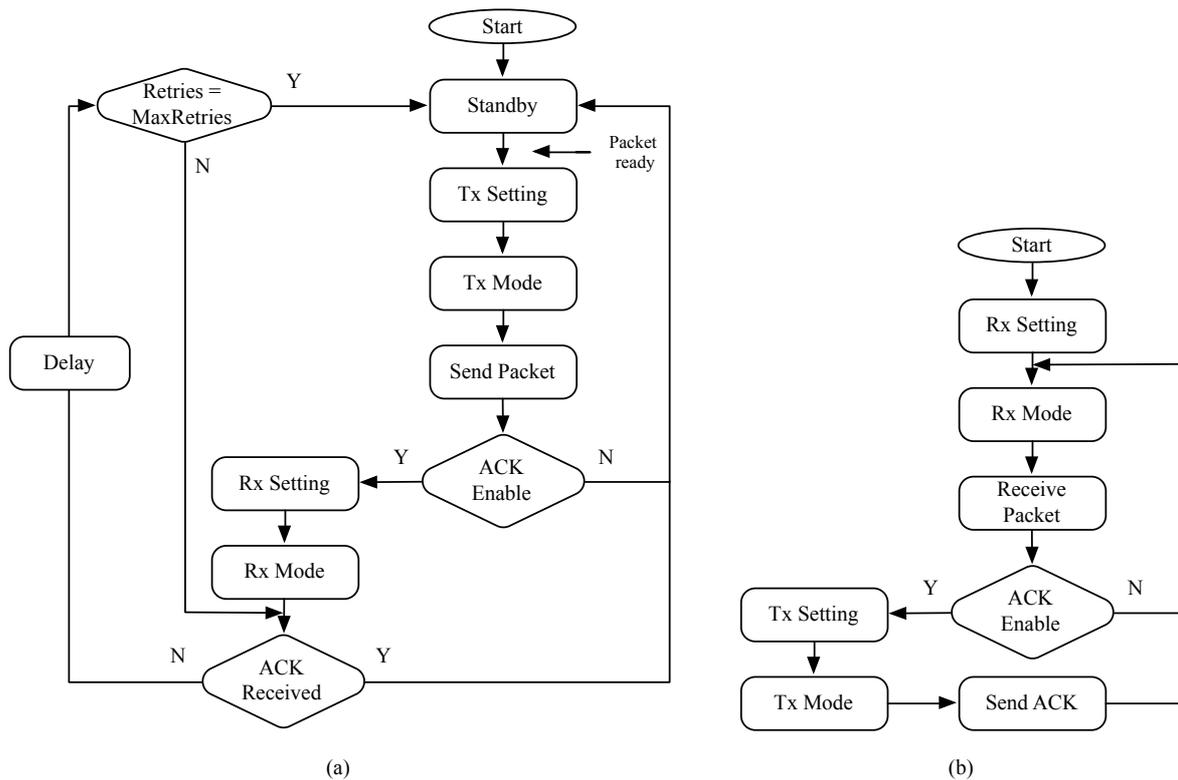


Figure 3.11: Organigramme de communication (a) Mode Tx (b) Mode Rx

3.3.7 Module d'acquisition des données

L'analyse spectrale du signal de vibration est connue comme l'une des approches les plus communes pour la surveillance et le diagnostic des défauts dans les machines tournantes. La collecte du signal de vibration est la tâche principale du module d'acquisition avant le traitement par l'unité FFT. L'accéléromètre est la solution idéale pour accomplir cette tâche en raison de son faible coût, sa taille miniature et qui présente une bonne réponse en fréquence. Par conséquent, une

bonne sélection devrait être faite pour sélectionner l'accéléromètre qui répond aux besoins des nœuds capteurs sans fil concernant la consommation d'énergie, la taille et les caractéristiques requises pour assurer un bon diagnostic des défauts.

Le développement des accéléromètres micro-électromécaniques (MEMS) en tant que transducteur de vibrations est une évolution dans la surveillance des vibrations par rapport aux accéléromètres traditionnels tels que les accéléromètres piézoélectriques. Les accéléromètres MEMS offrent plusieurs avantages : ils sont plus petits, à consommation énergétique beaucoup plus faible et moins coûteux. En plus, l'accéléromètre MEMS numérique dispose d'un circuit de conditionnement de signal, d'un amplificateur intégré, d'un convertisseur analogique-numérique, et d'un MCU. Il fournit directement un signal numérique, mais présente encore certaines limitations, où il ne peut pas être utilisé à des températures élevées et il a un bruit plus élevé par rapport aux capteurs piézo-électriques.

En 2008, une étude a été menée par les auteurs de [167], où trois accéléromètres MEMS ont été testés dans un atelier industriel typique. Les résultats indiquent que ces capteurs sont de bons candidats pour les implémentations sans fil, mais un choix doit être effectué selon les spécifications de l'environnement de mesure. De plus, les capteurs MEMS doivent être équipés d'un blindage approprié pour résister dans des environnements difficiles similaire à un environnement industriel.

Dans ce travail, nous avons adopté l'accéléromètre MEMS « ADXL345 ». Il est caractérisé par une très faible consommation avec 23 μA en mode mesure et 0,1 μA en mode veille, une résolution de 13 bits à ± 16 g, et une fréquence d'échantillonnage maximale de 3,2 kHz. Les données numériques de sortie sont codées en complément à deux sur 16 bits. La transmission des données est assurée via une interface numérique commune telle que le « SPI » ou « I2C ». Ce module est ajouté au nœud capteur comme un « Pmod » (Peripheral module) à travers un des « GPIO » disponible sur la plateforme basys-3, et ce dans le but d'avoir une structure modulaire du nœud capteur réalisé.

3.4 Résultats d'implémentation matérielle des nœuds

Le circuit FPGA choisi devrait disposer de suffisamment de ressources logiques, de blocs mémoires et de slices DSP, tout en respectant les contraintes des nœuds capteurs sans fil. Le tableau 3.8 résume les résultats de synthèse et d'implémentation des deux architectures sur la plateforme Basys-3, du nœud capteur et du nœud sink.

Selon ces résultats, nous pouvons voir que le nœud capteur a besoin de 93% des blocs BRAM et moins de 40% des autres ressources, ce qui s'explique par le besoin du bloc FFT à la mémoire de stockage avant et après le traitement des données. Comme déjà cité dans la section 3.3.4, la configuration du bloc FFT en mode streaming nécessite l'utilisation d'un IP-core « AXI stream FIFO », qui est plus adapté au traitement en continu, mais par conséquent il occupe beaucoup de ressources FPGA, plus précisément les BRAMs.

En termes de consommation d'énergie, nous avons pu arriver à faire fonctionner les nœuds réalisés à un niveau acceptable de puissance par rapport aux différentes conceptions similaires citées précédemment dans le tableau tableau 3.3. Il convient donc de dire que la consommation d'énergie dépend de la complexité des nœuds réalisés qui eux-mêmes dépendent de l'application.

Tableau 3.8: Résultats de synthèse et d'implémentation des nœuds

		Nœud capteur		Nœud Sink	
Ressources	Disponible	Utilisé	Utilisation (%)	Utilisé	Utilisation (%)
Slice LUT	20800	7646	36,76	2900	13,94
Bascules	41600	9334	22,44	2893	6,96
Bloc RAM	50	46,5	93	17	34
Slices DSP	90	36	40	-	-
Blocs IO	106	16	15,09	10	9,43
Puissance totale estimée sur puce		326 mW		230 mW	
Fréquence Max. du MicroBlaze	100 MHz				

3.5 Conclusion

Le développement des nœuds capteurs sans fil basés sur des circuits FPGAs est un axe de recherche futur. Cet axe a réussi à attirer l'attention des développeurs en raison des avantages apportés par les nouveaux circuits FPGAs qui peuvent diminuer l'écart entre les nœuds commerciaux et les nœuds de recherche en termes de capacités de traitement et d'énergie consommée, en plus de la grande flexibilité qu'il donnera aux concepteurs pour améliorer les architectures matérielles personnalisées. C'est le cas par exemple des FPGAs comme l'Artix de Xilinx ou bien Arria d'Altera. Ces circuits regroupent dans une seule puce des blocs spécialisés puissants avec une faible consommation d'énergie statique qui ne dépasse pas les 200 mW. On peut aussi citer la gamme Igloo d'Actel qui semble prometteuse car elles sont complètement optimisées pour les applications de faible puissance.

Dans ce chapitre, nous avons proposé des architectures pour des nœuds capteurs et pour un nœud sink à base d'un FPGA de type Artix. Cette architecture combine des IP-cores prêts à l'emploi, qui sont disponibles dans la bibliothèque des IP-cores de Vivado comme le MicroBlaze, la FFT, le « AXI streaming FIFO », ainsi que des IP-cores réalisés comme les blocs de cryptage et de décryptage. Ces derniers, qui sont conçus spécialement pour répondre aux besoins des nœuds sans fil en terme de sécurité tout en respectant les contraintes de la consommation d'énergie et la capacité de traitement.

Des résultats de simulation et d'implémentation ont été présentés en ciblant la plateforme : Digilent Basys-3. Les résultats obtenus ont montré l'adéquation des nœuds réalisés avec les exigences requises dans les RCSFs. Des tests pratiques dans une application industrielle seront présentés dans le prochain chapitre.

CHAPITRE 4 :

EVALUATION DES PERFORMANCES ET DE DIAGNOSTIC DES DEFAUTS DANS UNE APPLICATION INDUSTRIELLE

4.1 Introduction

Après avoir décrit l'architecture des nœuds proposés dans cette thèse, nous allons exposer dans ce chapitre, les résultats d'évaluation des performances, à savoir, le temps de réponse des différents blocs et modules constituant le nœud, la consommation d'énergie et la synchronisation temporelle des nœuds capteurs. Dans une deuxième partie, nous présenterons des tests pratiques réalisés pour un scénario d'application industrielle. Pour cela, nous avons choisi l'application de surveillance de l'état des machines industrielles, qui est l'une des applications des RCSFs les plus importantes en raison de ses besoins intrinsèques pour des solutions de traitements fiables, précises et sécurisées. L'intérêt de cette partie est de présenter le mode de fonctionnement de la solution proposée et de juger notre système dans différentes conditions de mesures, ce qui permettra de montrer son aptitude dans des applications ayant une pertinence industrielle.

4.2 Evaluation des performances

Cette partie présente l'évaluation des performances que nous avons effectuées sur les nœuds sans fil développés. Nous avons évalué d'abord l'opération d'acquisition de données par les nœuds capteurs à des taux d'échantillonnage élevés. Ensuite, on a effectué des mesures des temps de réponse nécessaires pour qu'un nœud capteur puisse accomplir les différentes tâches requises, allant de l'acquisition des données jusqu'à la transmission vers le nœud sink. De plus, nous avons évalué la consommation d'énergie du nœud capteur, et effectué ainsi, une estimation de sa durée de vie. Enfin, nous avons mesuré l'erreur de synchronisation temporelle entre deux nœuds capteurs appartenant au même réseau.

4.2.1 Acquisition de données à des taux d'échantillonnage élevés

La technologie des RCSFs a été appliquée avec succès dans de nombreux domaines tels que la surveillance de l'environnement, la surveillance de l'état des structures et la surveillance de la température dans la distribution des produits. Cependant, les dispositifs industriels imposent des exigences supplémentaires sur les RCSFs industriels (IWSN), tels qu'un taux d'échantillonnage plus élevé, des débits de transmission de données plus rapides et une plus grande fiabilité.

Un taux d'échantillonnage plus élevé permet d'obtenir une acquisition de données à haut débit et en temps réel, ce qui représente un défi dans les RCSFs en raison des ressources limitées du réseau, y compris la source d'énergie et la bande passante de communication.

Dans les systèmes de surveillance de l'état des machines industrielles pour la détection précoce des défauts et la maintenance prédictive, la plupart des applications mesurent des signaux acoustiques ou des vibrations de structures nécessitant des taux d'échantillonnage de plusieurs milliers de Hertz. Comme dans le cas de la surveillance des défauts de roulement décrite dans [68], où les auteurs proposent de calculer une moyenne quadratique (RMS) dans le domaine temporel, ou de détecter et de classifier les pics dans le spectre de fréquence. Généralement, les grandeurs physiques surveillées dans ce type d'applications exigent des fréquences d'échantillonnage allant de 5 à 40 kHz [168][169]. Une approche similaire est rapportée par [44] pour la surveillance des machines tournantes, où une acquisition de données des vibrations est obtenue avec des taux d'échantillonnage supérieurs à 100 kHz.

Pour détecter d'autres défauts tels que l'excentricité dynamique [170], les courts-circuits inter-spines [171], ou le déséquilibre de masse [2], des taux d'échantillonnage allant jusqu'à 10 kHz sont toujours requis, car la surveillance des moteurs à induction est moins exigeante du fait que les grandeurs physiques mesurées sont situées à proximité de la fréquence du réseau électrique (par exemple 50 Hz).

Afin d'évaluer les performances d'acquisition du nœud capteur sans fil réalisé pour des signaux à haute fréquence et un taux d'échantillonnage élevé, nous avons utilisé l'instrument multifonctions « Analog Discovery 2 » de Digilent pour générer quatre signaux sinusoïdaux avec différentes fréquences (500 Hz, 1 kHz, 5 kHz et 10 kHz) et avec la même amplitude (1 V).

Le signal analogique issu du générateur doit être converti en numérique. Pour cela, le nœud sans fil est équipé d'un convertisseur analogique-numérique de 12 bits de type ADCS7476MSPS. Les entrées disposent d'un filtre anti-repliement de type « Sallen key », avec un taux d'échantillonnage maximal égal à un million d'échantillons par seconde et qui peut être piloté facilement par le biais d'une liaison SPI.

Le taux d'échantillonnage du nœud est fixé à 50 kHz. Une fois l'opération d'acquisition terminée, une analyse spectrale (une FFT de 1024-points) des données collectées est effectuée dans le bloc FFT du nœud capteur. A l'issue de cette étape, les résultats du traitement ont été transmis à un PC via le nœud sink dans le but de tracer ces résultats en utilisant le logiciel Matlab. Les résultats obtenus sont représentés sur la figure 4.1. Ces résultats montrent que l'erreur relative entre la fréquence détectée par le nœud capteur et la fréquence réelle du signal généré est relativement faible et ne dépasse pas 2,5%. Cette erreur peut être réduite à 0,1% pour une fréquence de signal de 10 kHz.

4.2.2 Temps de réponse des tâches réalisées par le nœud capteur

Plusieurs mesures ont été effectuées dans le but de déterminer le temps requis pour les différents blocs et modules qui constituent le nœud capteur réalisé. Cette étape est nécessaire afin d'estimer par la suite la durée de vie du nœud. Dans ce test, le module accéléromètre « ADXL345 » est configuré à une fréquence d'échantillonnage de 3200 Hz, les données acquises sont ensuite transmises au bloc de traitement FFT (1024 points). Ces dernières sont cryptées par le bloc de cryptage AES avant d'être transmises via le module nRF24L01+, avec un débit maximal de transmission de 2 Mbps.

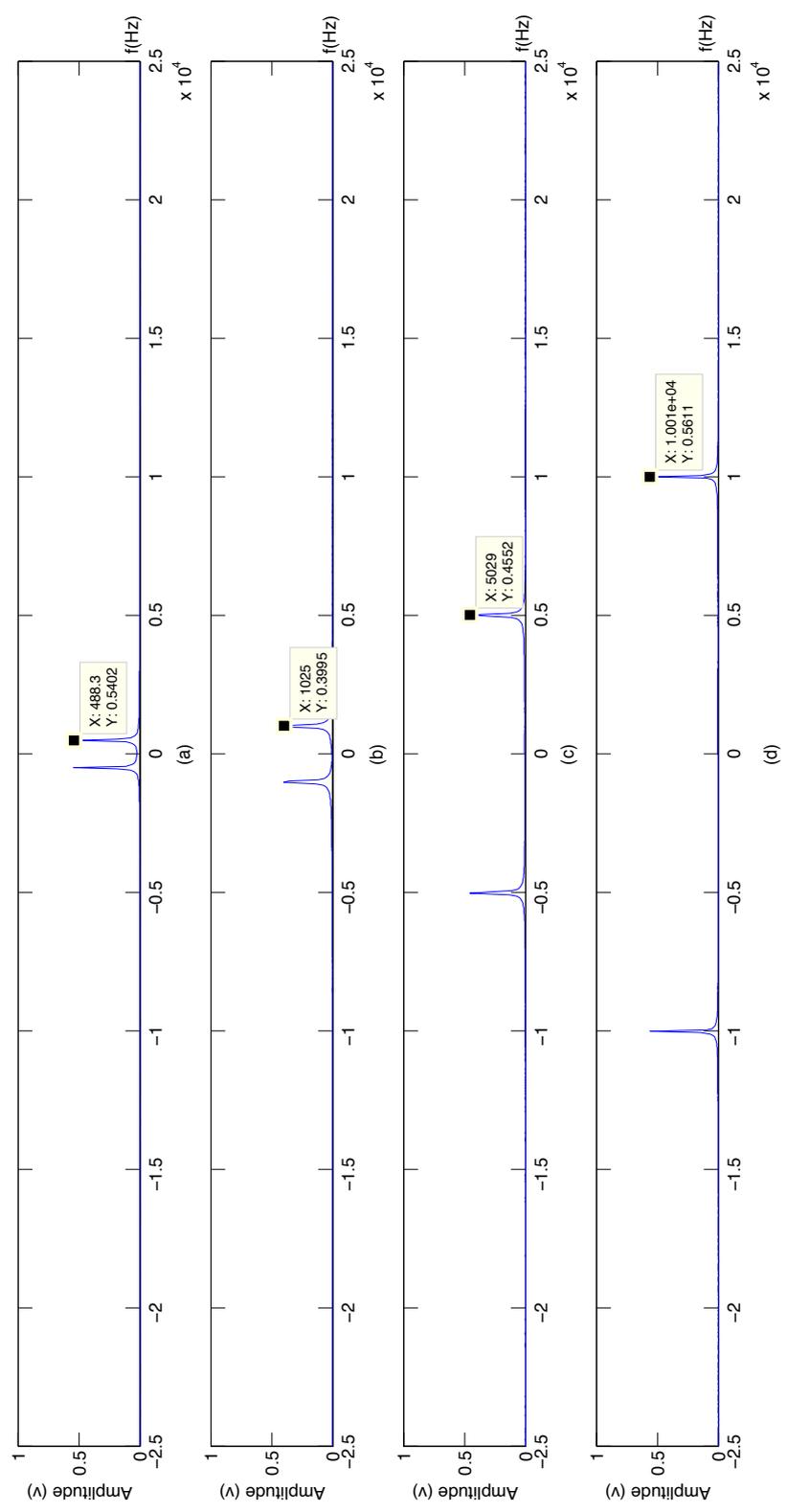


Figure 4.1: Représentation spectrale pour différentes fréquences du signal d'entrée (a) 500 Hz (b) 1kHz (c) 5 kHz (d) 10 kHz

Toutes ces opérations sont gérées par le processeur MicroBlaze. Ce dernier commande un signal qui indique le début et la fin de chaque tâche. Ce signal est affecté à une broche de sortie dans l'un des Pmods (Peripheral Module) disponibles sur la carte FPGA. Un analyseur logique est utilisé pour la mesure des différents temps de réponse. Les résultats sont présentés sur la figure 4.2. Au vu de ces résultats, nous pouvons constater que le temps total nécessaire pour un traitement complet dans le nœud est inférieur à 3 s, pouvant varier en fonction de la fréquence d'échantillonnage, ce qui indique que le nœud proposé convient à des mesures périodiques. En outre, le temps requis pour le cryptage est d'environ 605 μ s, ce qui est très négligeable par rapport au temps requis pour les autres tâches, Ceci est dû au choix d'une conception hardware du bloc de sécurité (un IP-core) qui accélère considérablement le délai nécessaire pour le cryptage des données.

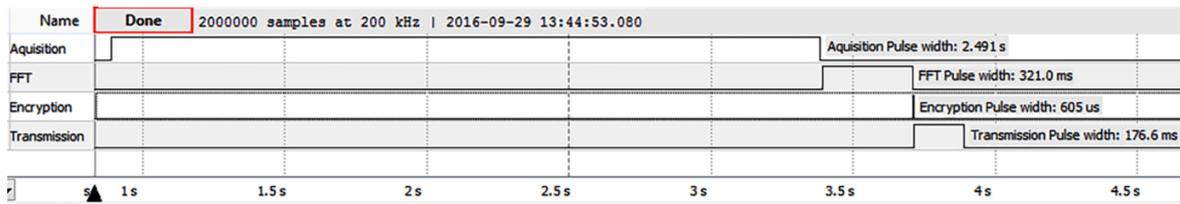


Figure 4.2: Temps de réponse des tâches réalisées par le nœud capteur

4.2.3 Consommation d'énergie du nœud capteur sans fil

La consommation moyenne d'énergie du nœud capteur est estimée en s'appuyant sur les données techniques du module capteur et du module RF, ainsi que sur les résultats de l'implémentation sur FPGA présentés auparavant (voir chapitre 3).

Pour étendre la durée de fonctionnement du système de surveillance, nous avons pourvu le nœud capteur par des composants de faible puissance. En outre, les deux modules d'acquisition et de communication disposent du mode de veille. Afin d'estimer cette consommation, nous avons supposé que le nœud est alimenté par une batterie à courant continu de 3,6 V. Les résultats du courant moyen obtenus ainsi que la puissance consommée sont indiqués dans le tableau 4.1. Étant donné que la capacité de la batterie au lithium sélectionnée est de 1000 mAh, la durée de vie du fonctionnement du nœud peut être évaluée et les résultats sont également

donnés dans le tableau 4.1.

Tableau 4.1: Résultats de la consommation d'énergie du nœud pour les deux modes de fonctionnement

Mode de fonctionnement	Courant (mA)	Puissance (mW)	Durée de vie (h)
Mode Opérationnel	101.873	366.74	9.81
Mode Veille	20.28	73.022	49.30

Dans le mode opérationnel, le nœud peut fonctionner en continu pendant environ 9 heures. Dans ce cas, le nœud capteur effectue périodiquement une mesure toutes les 3 s, ce qui est très élevé pour la majorité des applications de surveillance dans le milieu industriel. Lorsque tous les modules entrent en mode veille, la consommation de courant n'est que de 20,28 mA, ce qui augmente la durée de vie à plus de 49 h. Par conséquent, ce mode de veille contribue à prolonger cette durée. En fait, elle dépend principalement du plan de surveillance. Par exemple, avec une surveillance de périodicité de 1 minute (3 s en mode opérationnel et 57 s en veille) de collecte de données, elle passe de 9 h à plus de 40 h (le deuxième point sur le graphe de la figure 4.3). La limite de 49 h peut être aussi atteinte avec une périodicité d'une heure.

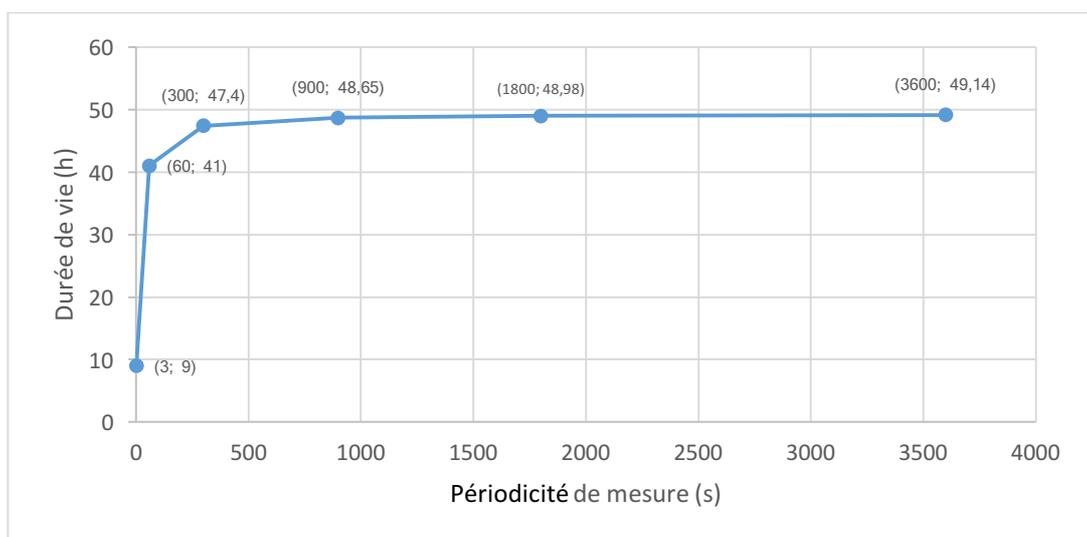


Figure 4.3: Durée de vie du nœud en fonction de la périodicité de mesure

4.2.4 Synchronisation temporelle des nœuds capteurs

Pour de nombreuses applications de surveillance des machines, il est important que l'acquisition de données sur les différents nœuds capteurs d'un RCSF s'exécute en même temps, dans un intervalle d'erreur acceptable. Cela devient plus difficile si les nœuds sont dispersés, et plus encore si le canal de communication subit des retards variables ou s'il n'est pas fiable.

Comprendre les retards dans la transmission radio aide à développer et à évaluer les systèmes de synchronisation dans les RCSFs, ce qui permet de concevoir des protocoles qui éliminent ces retards. Ganeriwal et al. [172] ainsi que Maroti et al. [173] ont identifié six éléments de retard dans la transmission des paquets.

- Temps d'envoi : Lorsqu'un nœud décide de transmettre un paquet, il est programmé comme une tâche dans un nœud capteur typique. Le temps nécessaire pour créer, assembler réellement le paquet dans la couche d'application et émettre la demande d'envoi sur la couche MAC est défini comme un temps d'envoi. C'est un temps indéterminé et peut atteindre les centaines de millisecondes.
- Temps d'accès : Le temps pris à partir de la réception du paquet par la couche MAC pour accéder au support physique (canal d'émission). Ce délai est spécifique aux réseaux sans fil résultant de la propriété de support de transmission de paquets.
- Temps de transmission : C'est le temps qu'il faut pour que le nœud émetteur envoie un paquet. Ce délai est principalement de nature déterministe et peut être estimé en utilisant la taille du paquet et le débit de transmission.
- Temps de propagation : Il s'agit du temps réel nécessaire pour que le paquet traverse le lien sans fil de l'émetteur au récepteur. La valeur de ce délai est très déterministe dans un RCSF et elle dépend uniquement de la distance entre les deux nœuds.
- Temps de réception : Le temps nécessaire pour que le destinataire reçoive le message et qu'il soit transmis à la couche MAC.

- **Durée de réception** : C'est la durée nécessaire à la couche MAC pour construire le message entrant et le transmettre à la couche d'application où il est décodé et pour notifier la demande du destinataire.

Il existe plusieurs protocoles de synchronisation temporelle en raison de différentes exigences, telles que la précision ou le degré de mobilité. Dans le but d'assurer une acquisition synchrone entre les différents nœuds du réseau, le nœud sink est chargé d'ordonner le lancement simultané de l'acquisition sur tous les nœuds cibles. Cette solution est améliorée à l'aide d'un protocole de communication.

Notre solution adoptée est représentée à la figure 4.4. Cet organigramme montre les différentes étapes impliquées dans le procédé de synchronisation des nœuds capteurs. Initialement, le nœud sink est configuré en mode de transmission (Tx) et les nœuds capteurs sont configurés en mode de réception (Rx). Ensuite, les nœuds capteurs attendent un paquet spécifique, arrivant du nœud sink qui démarre l'opération d'acquisition.

Une fois que le nœud capteur reçoit ce paquet, les données utiles sont stockées dans le « Rx FIFO ». En plus, une interruption est générée, et le bit « Rx_Data_Ready » du registre d'état du module nRF24L01+ est mis à « 1 ». Le bit « Rx_Pipe_Number » du registre d'état indique le flux de données sur lequel la charge utile a été reçue. En d'autres termes, un accès direct de la couche physique à la couche d'application aura lieu, indiquant au MicroBlaze de commencer le processus d'acquisition. Simultanément, un signal de front montant est envoyé à une broche d'E/S dans l'un des Pmods sur la carte FPGA, utile pour l'évaluation qui sera présentée plus tard. Ensuite, les nœuds capteurs basculent en mode Tx, et en même temps, le nœud sink bascule vers le mode « Multiceiver » et attend le paquet d'accusé de réception (ACK) provenant des nœuds capteurs, qui est transmis automatiquement par le module RF. Si un paquet d'accusé de réception est perdu, le nœud sink néglige les données reçues provenant de l'ensemble des nœuds et une nouvelle demande est lancée pour entamer une nouvelle tentative de synchronisation. Dans le cas où le nœud sink reçoit les paquets ACK de tous les nœuds capteurs constituant le RCSF, elle commence à recevoir les données des

nœuds capteurs. Une fois la transmission terminée, le nœud sink repasse en mode de transmission (Tx) et les nœuds capteurs en mode de réception (Rx) pour qu'ils se préparent à la prochaine acquisition.

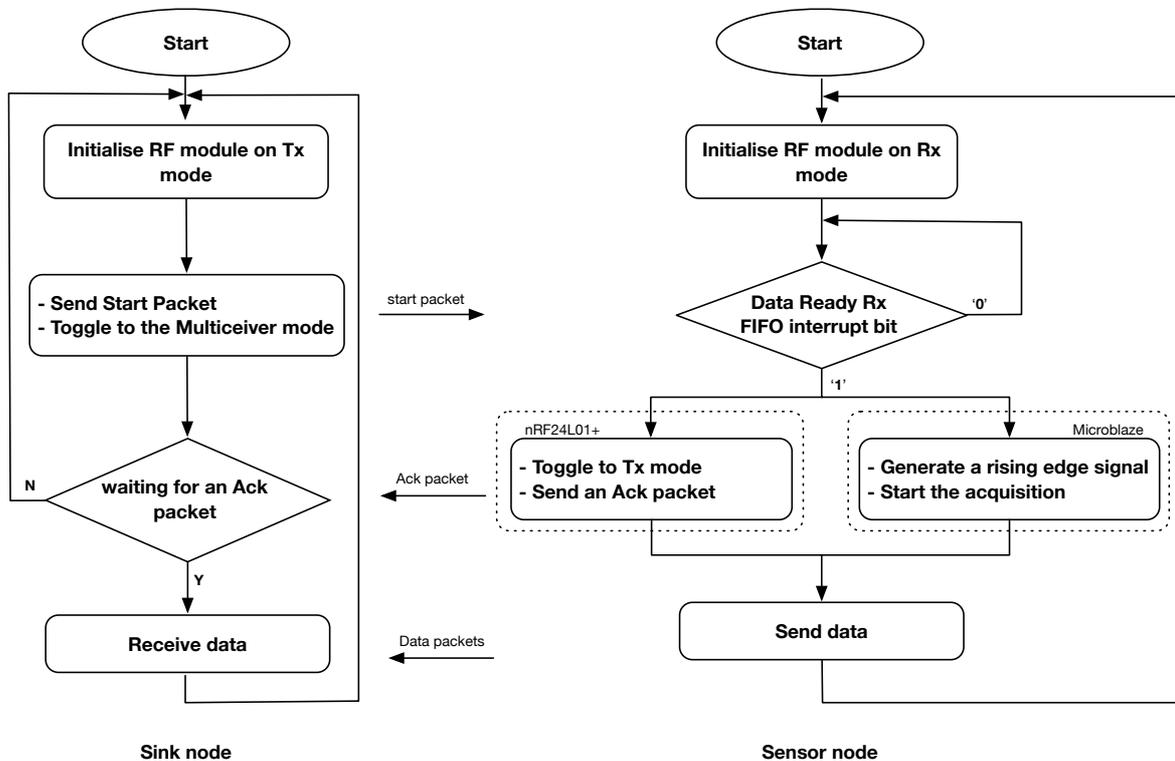


Figure 4.4: Organigramme du procédé de synchronisation

L'évaluation de la synchronisation d'acquisition entre les nœuds a été effectuée par des tests pratiques dans le but de voir l'influence de l'environnement.

Dans une première série d'expériences, tous les nœuds sont placés en visibilité directe. Nous avons effectué une série de mesures à l'aide d'un oscilloscope. Ce dernier visualise les signaux des fronts montants au niveau des nœuds capteurs. Chaque front montant indique le début de l'acquisition de données de son nœud. Le délai entre deux fronts montants nous donne la valeur mesurée de l'erreur commise dans le procédé de synchronisation dans le cas de deux nœuds capteurs (Figure 4.5).

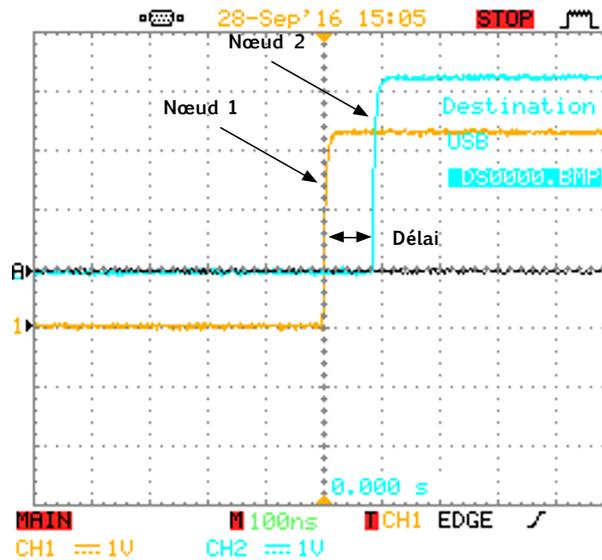
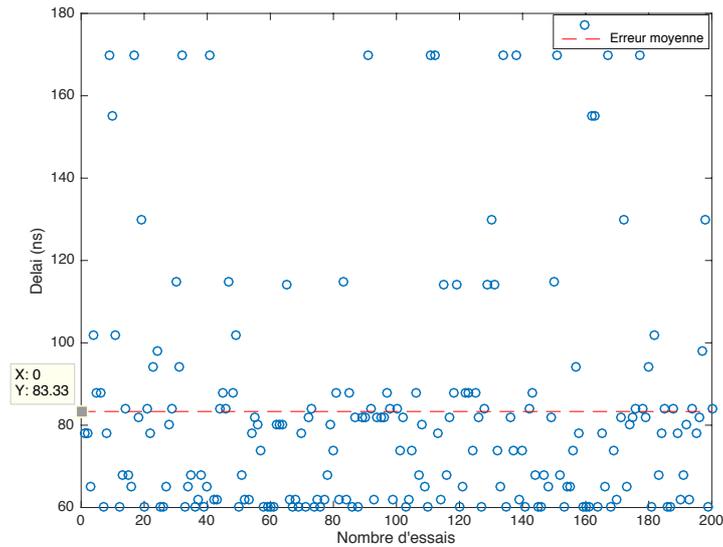


Figure 4.5: Erreur mesurée commise dans le procédé de synchronisation

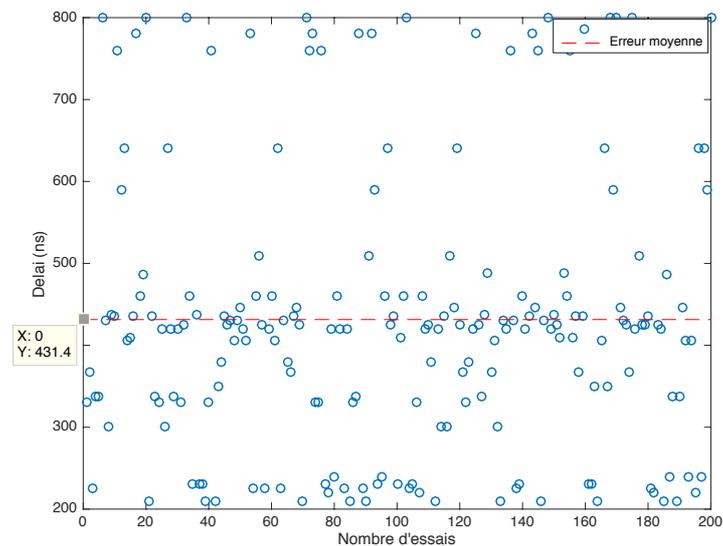
Les résultats des tests représentés sur la figure 4.6.a, montrent que l'erreur est limitée entre 60 ns et 170 ns avec une erreur moyenne de 83,33 ns.

Pour une deuxième série d'expériences, nous avons appliqué un environnement similaire à celui d'un environnement industriel où la présence d'obstacles peut affecter la qualité de communication et introduit une dispersion temporelle et des échos causés par les retards de propagation par les multi-trajets, qui, à leur tour, affecteront la synchronisation des données.

Les résultats des tests représentés sur la figure 4.6.b, montrent que l'erreur peut être considérablement augmentée et fluctue entre 210 ns et 800 ns avec une erreur moyenne de 431,42 ns. Malgré cela, les résultats obtenus restent acceptables pour plusieurs applications de surveillance [19], même dans des conditions défavorables.



(a)



(b)

Figure 4.6: Résultats des tests d'erreurs de synchronisation mesurées (a) 1^{ère} expérience (visibilité directe) (b) 2^{ème} expérience (présence d'obstacles)

4.3 Application pour la surveillance vibratoire des machines tournantes

Dans un monde où la compétitivité industrielle est de plus en plus accrue, la production à temps est un facteur important pour de nombreuses industries. Les coûts élevés impliqués dans les retards de production en raison de l'échec du matériel industriel sont donc indésirables. Plusieurs méthodologies de maintenance

sont utilisées en pratique pour assurer la bonne fonctionnalité des machines dans l'industrie. Ces approches sont classées comme suit : une maintenance préventive, prédictive et conditionnelle [174].

Dans le but d'exploiter au maximum les équipements et les machines, les travaux de maintenance ne sont lancés que lorsqu'une panne apparaît et par conséquent conduit à un arrêt de production qui peut être important et coûteux financièrement, en plus de l'exposition au danger pour les personnes en contact avec ces équipements. Cette approche est en premier lieu corrective et est effectuée après la défaillance. La maintenance préventive surmonte les faiblesses de la maintenance corrective en introduisant des travaux de réparation à des intervalles de temps prédéterminés. Les travaux de maintenance sont effectués à des intervalles de temps définis en fonction des heures de fonctionnement de la machine. Cependant, cela ne permet pas une utilisation efficace de la machine et entraîne souvent des arrêts de maintenance inutiles.

La maintenance conditionnelle est une stratégie prometteuse qui est basée sur une surveillance continue de l'état de fonctionnement et vise à remédier aux causes profondes du problème afin d'éviter les échecs futurs [175]. Par conséquent, l'avantage principal associé à la méthode de surveillance des conditions réside non seulement dans les travaux de maintenance bien planifiés, mais permet également de corriger les causes profondes des défaillances par l'intégration dans le processus de décision des informations et des indicateurs sur l'état courant du système. Dans une telle approche, Il existe différentes techniques d'analyse telles que l'analyse vibratoire, l'émission acoustique, la thermographie, l'analyse des huiles et des lubrifiants, la variation de résistance dans un circuit électrique, ...etc. Le choix de l'indicateur dépend du type de machine à étudier et du type de défaillance que l'on souhaite détecter. Pour les machines tournantes, un indicateur de type vibratoire permet de détecter la plupart des défauts [176].

Dans ce contexte, nous avons exploité un RCSF, composé des nœuds développés, dans une application basée sur une technique d'analyse vibratoire d'une machine tournante, en raison de son utilité pratique, sa capacité de traitement d'une grande

quantité de données et la nécessité d'une analyse intensive en calcul dans une telle application.

4.3.1 Surveillance par analyse vibratoire

Cette méthode est très utilisée pour la surveillance des machines tournantes [174](Figure 4.7), car la réponse vibratoire d'une machine est sensible à tout changement de paramètre structurel ou de processus [177].

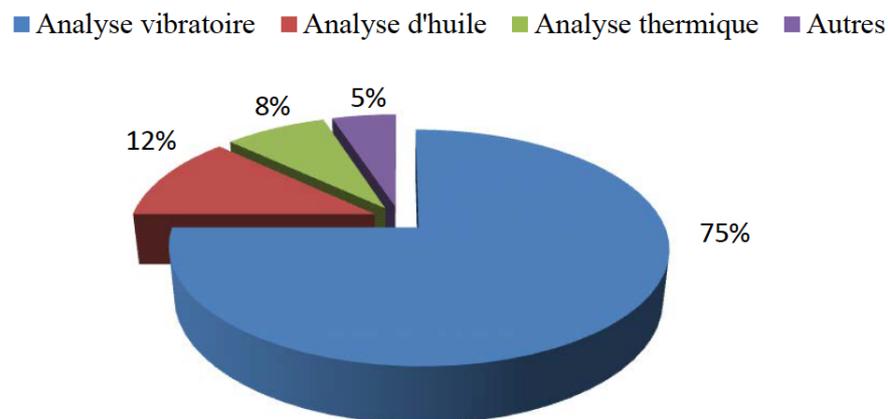


Figure 4.7: Les différentes méthodes d'analyse

Elle est basée sur l'analyse du comportement vibratoire des machines en fonctionnement ainsi que de ses caractéristiques dans le temps, la fréquence ou temps-fréquence (Figure 4.8) et ce dans le but d'établir un diagnostic sur leur état de marche. Ceci pourrait indiquer des dommages ou de la dégradation [178]. Les données de vibration sont soit obtenues par déplacement, par vitesse ou par accélération.

4.3.2 Problèmes fréquemment diagnostiqués dans les machines tournantes

En général, toutes les machines tournantes produisent des vibrations lorsqu'elles sont utilisées. Ces vibrations sont généralement fonction de la dynamique de la machine et des paramètres d'installation tels que l'alignement, l'équilibre, etc. Le diagnostic de l'état d'une machine n'est possible que si l'on connaît les images

vibratoires associées à chaque défaut susceptible d'affecter la machine considérée. Par conséquent, en mesurant la vibration et en la reliant aux paramètres de dynamique, d'alignement et d'équilibre, l'état de la machine ainsi que celui de ses pièces associées peut être déterminé. Les défauts de machines qui sont couramment diagnostiqués à l'aide de données de vibration incluent le déséquilibre, le désalignement, la détérioration structurelle, les barres de rotor brisées, le frein de roulement et le frottement, le désalignement et l'usure de la ceinture, le désalignement et l'usure des engrenages, les barres de rotor lâches et un entrefer inégal. La plupart des problèmes dans les machines tournantes sont dus au déséquilibre mécanique, au désalignement et aux problèmes de fixations qui sont mieux détectés par l'analyse des vibrations. Nous avons focalisé notre contribution sur un des défauts les plus courants, à savoir le déséquilibre massique des rotors souvent appelée (défaut balourd).

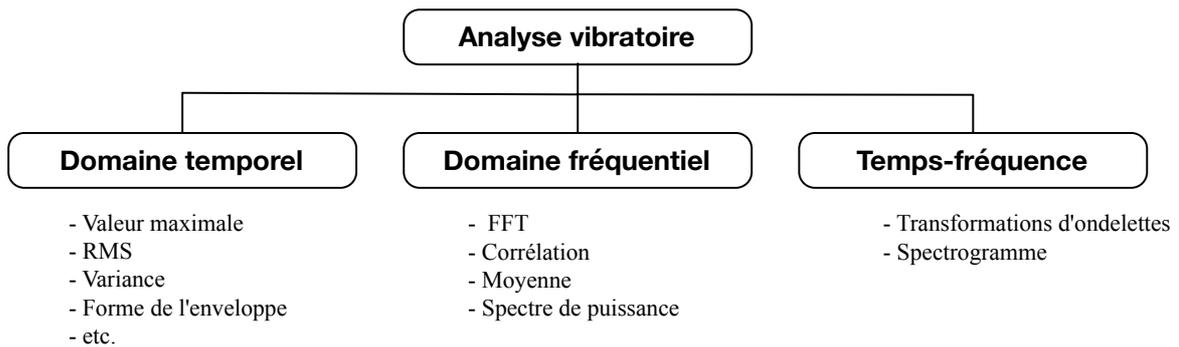


Figure 4.8: Les méthodes d'extraction des caractéristiques basées sur les vibrations

4.3.3 Les défauts balourds

Le balourd est l'une des principales causes de vibrations sur les machines tournantes. Il est généralement dû à une répartition irrégulière des masses autour de l'axe du rotor. Quel que soit les précautions apportées durant la construction des machines, il est impossible de faire correspondre l'axe de rotation avec le centre de gravité de chaque tranche élémentaire du rotor. Il en résulte par conséquent, des efforts centrifuges qui déforment l'arbre en rotation et engendrent des déséquilibres [179]. Ces efforts se traduisent par des vibrations liées à la fréquence de rotation

puisque la force centrifuge est proportionnelle à la masse du balourd, au rayon du rotor et à la vitesse de rotation de l'arbre (Figure 4.9) [180].

Ces déséquilibres peuvent survenir dans le cas d'un défaut d'usinage et d'assemblage des rotors. Les rotors peuvent aussi se déformer à cause de l'échauffement ainsi que de la dilatation thermique des matériaux durant la phase de fonctionnement. Ces déséquilibres peuvent aussi être d'origine mécanique comme dans le cas des pertes d'ailettes et d'aubes, d'érosion, d'encrassement, ... etc. C'est ainsi qu'un défaut de balourd se révèle par :

- Une composante d'amplitude quelque peu élevée à la fréquence de rotation du rotor en direction radiale, très difficile à détecter à l'état précoce ;
- Un déphasage de 90° entre deux composantes correspondant à des points de mesures radiaux sur le même palier.

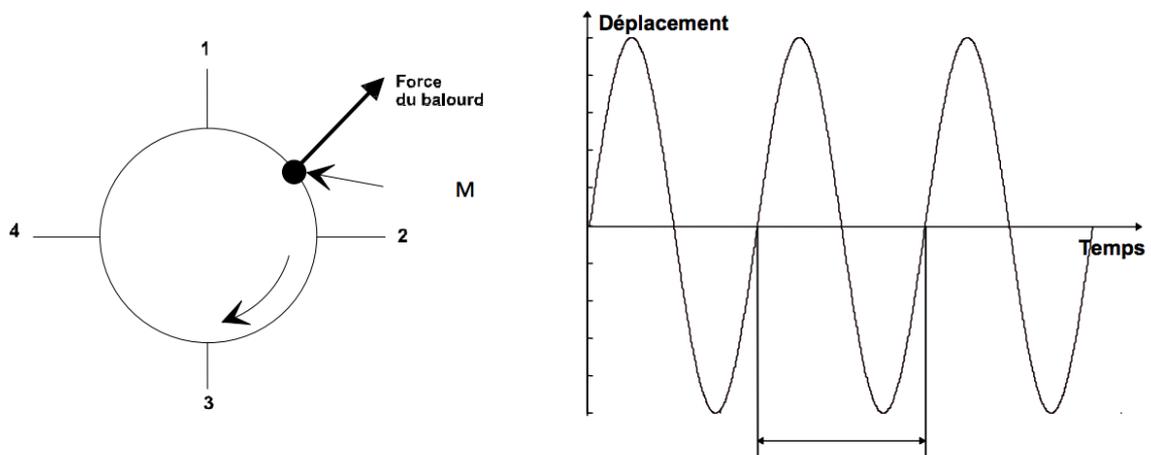


Figure 4.9: Signal vibratoire généré par le balourd

4.3.4 Description du banc d'essai

Afin de réaliser les tests expérimentaux, nous avons utilisé un banc d'essai constitué d'un disque monté sur un arbre supporté par deux paliers auto-aligneur de type P204 et accouplé avec un moteur asynchrone d'une vitesse maximale de 2765 tr/min. L'ensemble repose sur un cadre châssis. La commande de la vitesse

est assurée par un variateur de fréquence de type ALTIVAR312. L'ensemble du banc d'essai utilisé est montré sur la Figure 4.10. Les trous percés sur le disque sont utilisés pour la fixation de petites masselottes dans le but de matérialiser un déséquilibre balourd.

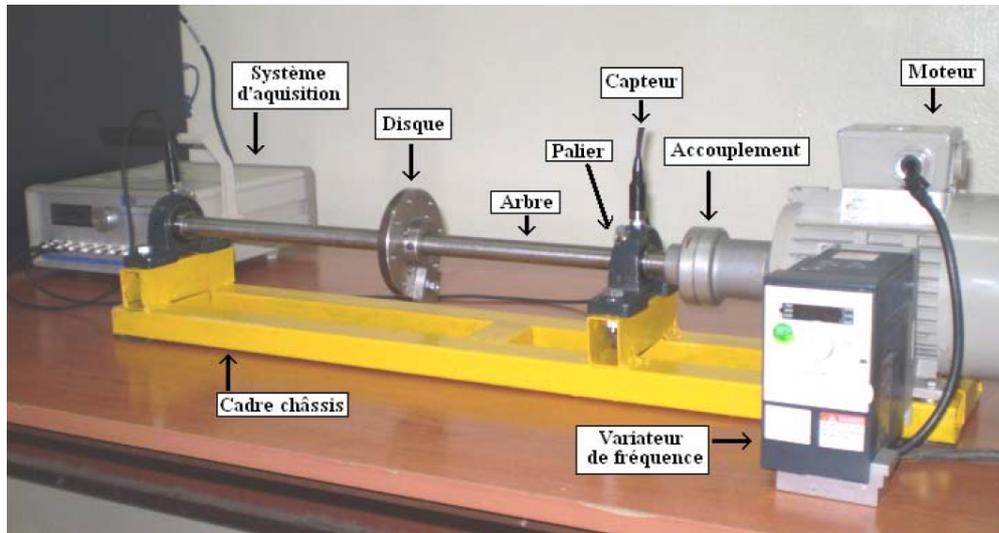


Figure 4.10: Composition du banc d'essai utilisé

L'emplacement du capteur ainsi que ces modes de fixation sont des paramètres qui influent considérablement sur les données acquises. Donc, le choix de l'implantation des points de mesure sur les machines est un facteur important pour la réussite de l'analyse vibratoire. En effet, les images vibratoires issues d'une analyse d'un phénomène mécanique peuvent être sensiblement différentes en fonction du point de mesure. On essaiera toujours de placer les accéléromètres au plus près de la source du défaut, en se rapprochant le plus possible des points de mesure des paliers, afin d'éviter toutes contributions extérieures et d'obtenir les images les plus fidèles des défauts mécaniques.

Pour chaque ligne d'arbre, les emplacements choisis doivent permettre :

- D'effectuer une mesure radiale horizontale, une mesure radiale verticale et une mesure axiale sur le palier du côté transmission (du côté de la charge maximale).
- De limiter au strict minimum le nombre de pièces assurant l'interface entre

l'élément mobile et le capteur.

- D'assurer l'isolement du capteur pour la prise de mesure sur les appareils électriques.

L'emplacement des points de mesure doit être propre, les surfaces de contact avec les capteurs doivent être également lisses, planes et perpendiculaires à la direction de la mesure. L'utilisation des câbles de liaison ne doit pas générer des contraintes mécaniques qui entraînent des parasites. Il faut donc les fixer de manière adéquate et éviter les longueurs inutiles. En générale, la mesure du spectre pour un défaut balourd est prise radialement, souvent dans la direction radiale horizontale comme indiqué sur la figure 4.11.

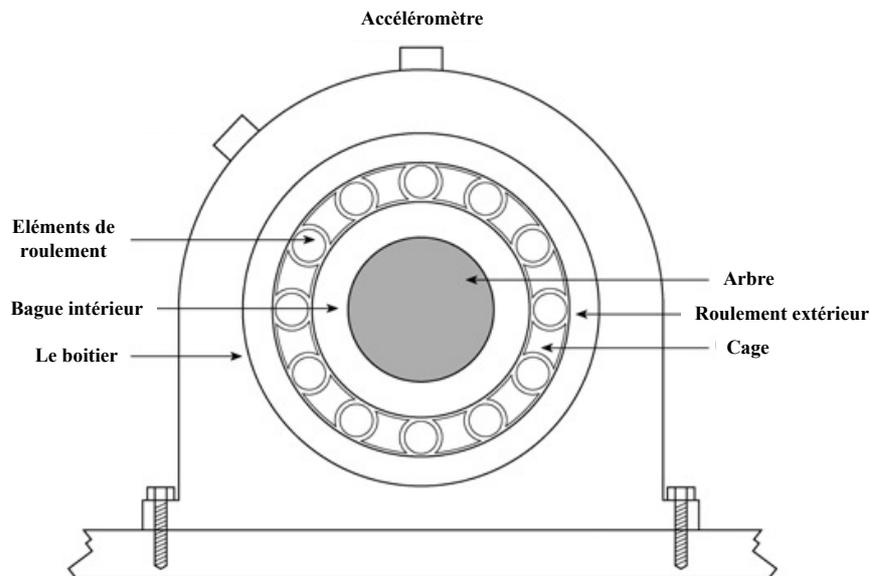


Figure 4.11: Emplacement de l'accéléromètre sur le palier

4.3.5 Tests pratiques de diagnostic de défaut balourd

Afin d'évaluer les nœuds proposés dans une application réelle, plusieurs tests pratiques ont été réalisés pour mesurer les signaux de vibration de la machine tournante, donnant ainsi la possibilité de statuer sur le diagnostic du défaut. Dans ces tests, nous avons créé un déséquilibre pour générer un défaut de balourd, ce qui provoque des niveaux élevés d'amplitudes de vibration à la fréquence de rotation de la machine.

Dans ce contexte, nous avons réalisé un RCSF composé de deux nœuds capteurs et un nœud sink, présentés précédemment et dont leurs modules capteurs sont fixés sur les deux paliers d'extrémités. Ensuite, ces nœuds capteurs effectuent les traitements FFT et cryptage AES. Enfin, les données traitées sont transmises au nœud sink connecté à un PC. Elle commence par le décryptage des données reçues, et puis les envoie vers un PC portable, où le logiciel Matlab est utilisé pour tracer les différents graphes. Le système global de surveillance réalisé est représenté sur la figure 4.12. Nous avons configuré l'accéléromètre pour deux fréquences d'échantillonnage ($F_{s1} = 400$ Hz et $F_{s2} = 3200$ Hz).

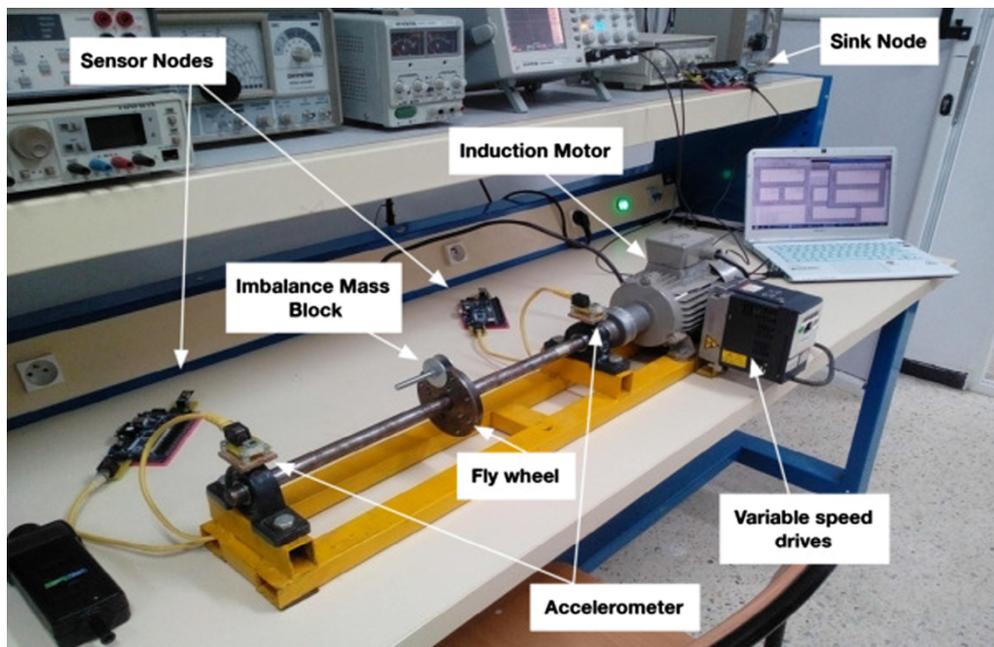


Figure 4.12: Système global de surveillance réalisé

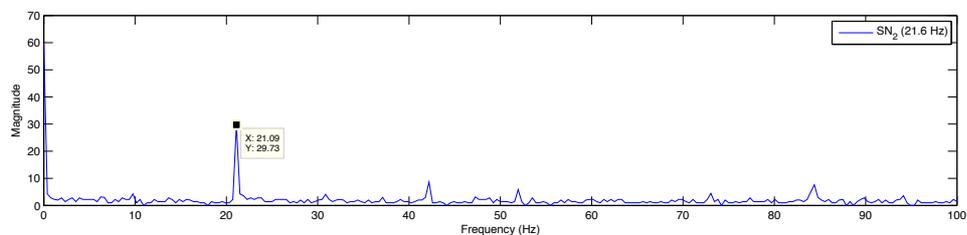
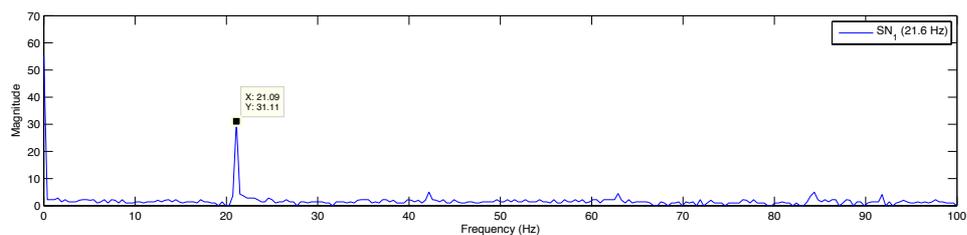
Les résultats sont présentés sur les figures 4.13 et 4.14, où les spectres de fréquences des signaux mesurés et envoyés par les deux nœuds capteurs sont représentés.

Dans l'exemple du résultat présenté sur la figure 4.13.a, la vitesse de rotation du moteur est mesurée par un stroboscope réglé sur 1296 tr/min, c'est l'équivalent d'une fréquence de 21,6 Hz. Une masse est fixée sur le disque pour générer un

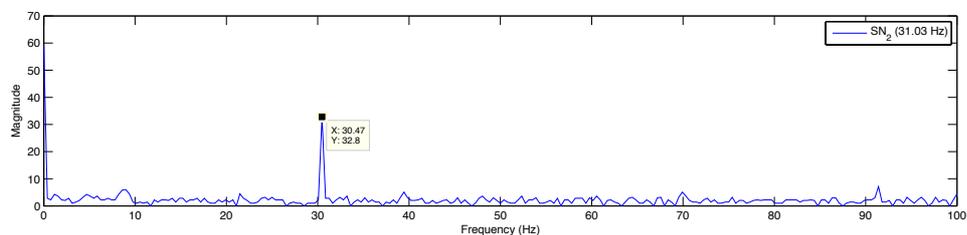
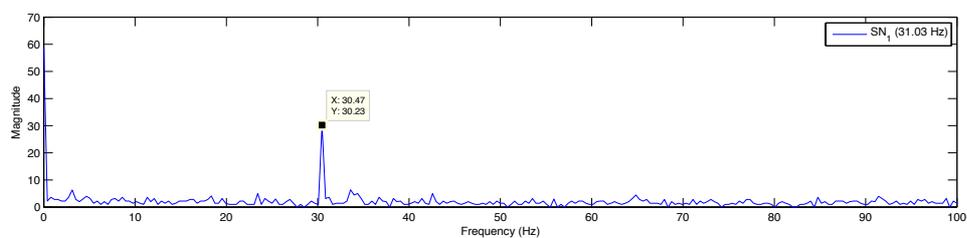
défaut balourd. À une fréquence d'échantillonnage de 400 Hz, les résultats provenant des deux nœuds indiquent la présence d'un pic à la fréquence de 21,09 Hz, ce qui donne une erreur de 2,36% par rapport à la fréquence de rotation réelle. Cette erreur est réduite à 1,29% en augmentant la fréquence d'échantillonnage à 3200 Hz. Sur la figure 4.13.b, la vitesse de rotation est réglée à 1862 tours/minute, les deux nœuds capteurs indiquant la présence d'un pic à la fréquence de 30,47 Hz avec une erreur de 1,83%. Il en est de même, pour le reste des résultats qui sont représentés sur le tableau 4.2. Ces résultats montrent que l'erreur peut être réduite chaque fois que la fréquence d'échantillonnage est augmentée (de 400 Hz à 3,2 kHz), en particulier pour une vitesse de rotation de 2430 tours/minute où l'erreur diminue de 1,65% à 0,32%, c'est-à-dire avec une réduction d'erreur de 80%. On peut également noter que pour chaque expérience, les deux nœuds capteurs donnent un pic à la même fréquence justifiant l'effet de synchronisation dans le RCSF réalisé.

Tableau 4.2: Résultat de diagnostic pour deux fréquences d'échantillonnage. SN1 et SN2 désignent respectivement le nœud capteur 1 et le nœud capteur 2.

Vitesse de rotation (tr/min)	$F_{s1} = 400 \text{ Hz}$		$F_{s2} = 3200 \text{ Hz}$	
	Fréquence mesurée (Hz) (SN1 , SN2)	Erreur relative (%)	Fréquence mesurée (Hz) (SN1 , SN2)	Erreur relative (%)
1296 (21,6 Hz)	21,09	2,36	21,88	1,29
1862 (31,0 Hz)	30,47	1,83	31,25	0,70
2430 (40,5 Hz)	39,83	1,65	40,63	0,32
2712 (45,2 Hz)	44,53	1,48	45,05	0,33



(a)



(b)

Figure 4.13: Spectre du signal mesuré par les nœuds capteurs et reçue par le sink. Fréquence de rotation : (a) 21,6 Hz ; (b) 31,01 Hz

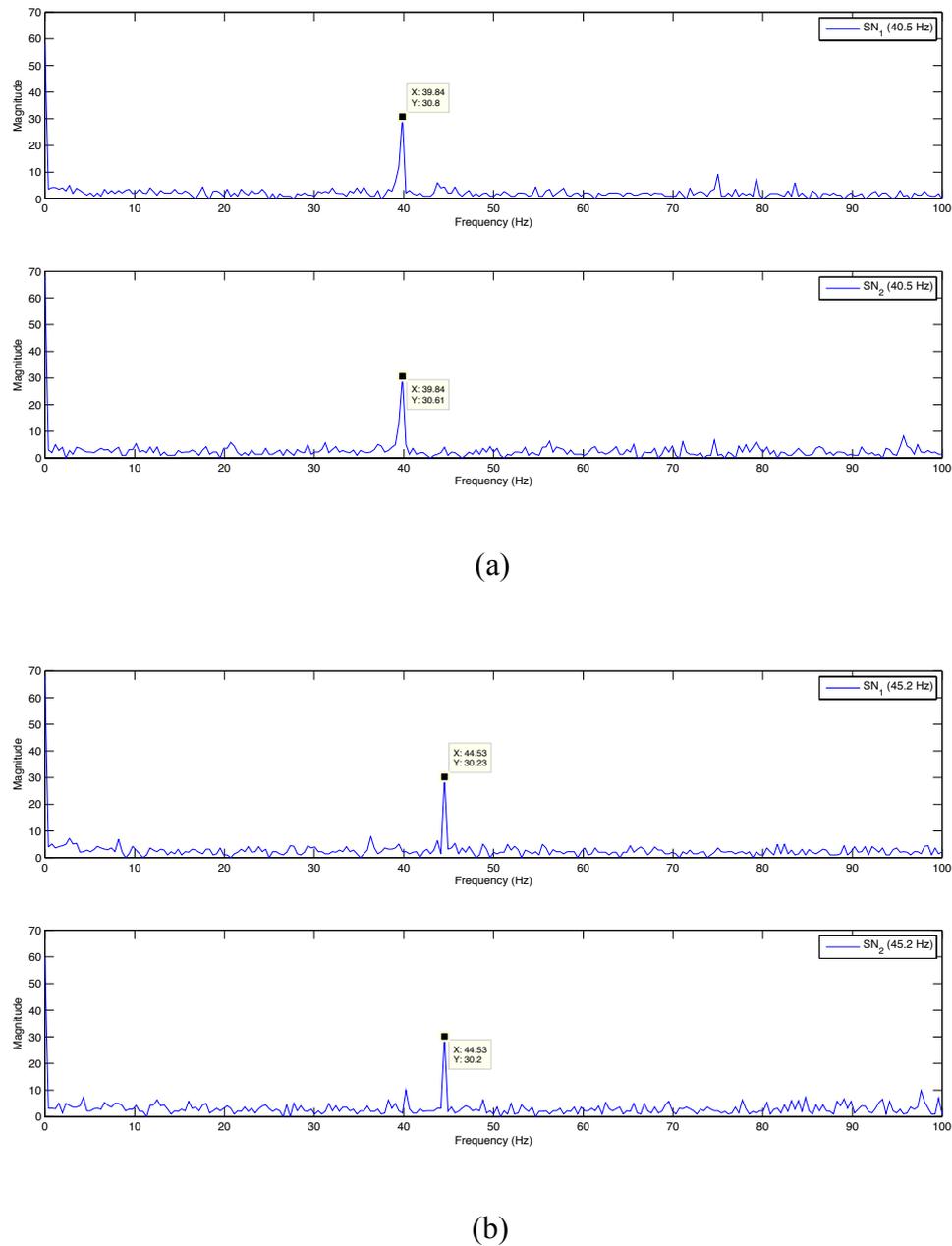


Figure 4.14: Spectre du signal mesuré par les nœuds capteurs et reçue par le sink. Fréquence de rotation : (a) 40,5 Hz ; (b) 45,2 Hz

4.4 Conclusion

L'objectif dans cette dernière partie du travail, était d'évaluer les performances des différents nœuds proposés, afin de répondre aux différentes exigences, comme le taux élevé d'échantillonnage et l'acquisition synchrone de haute précision appliqués dans une application de surveillance des vibrations mécaniques. Le nœud proposé,

à base d'une architecture conjointe matérielle/logicielle est conçu pour garantir un compromis entre la consommation d'énergie et les performances. Il supporte une fréquence d'échantillonnage élevée allant jusqu'à 10 kHz et permet l'acquisition des signaux de vibration des machines industrielles. Une méthode d'acquisition synchrone basée sur un protocole de communication gérée par le nœud coordinateur ou sink est proposée dans le but d'assurer une synchronisation temporelle de précision pour les différents nœuds capteurs. Les tests pratiques montrent que l'erreur moyenne obtenue varie selon le milieu des essais, ce qui influe considérablement sur les résultats, avec 83,33 ns obtenu dans un milieu ouvert et 431,42 ns dans un milieu déformé. Les nœuds réalisés sont validés par plusieurs tests de mesures et les résultats montrent que les nœuds proposés sont efficaces pour la surveillance des vibrations des machines industrielles.

CONCLUSION GENERALE ET PERSPECTIVES

Les nœuds capteurs sans fil qui sont les composants de base d'un RCSF sont généralement contraints par plusieurs considérations telles que la taille physique, le coût, la consommation énergétique et la capacité de traitement. Ces contraintes ainsi que le besoin d'avoir une longue durée de vie du réseau, limitent généralement la conception du nœud capteur de différentes façons. L'une d'entre elles consiste à utiliser des modules à faible consommation et peu coûteux, ce qui en conséquence, offre des performances limitées en transmission et en traitement. D'autre part, les applications de surveillance qui impliquent de grandes quantités de données et des algorithmes de traitement complexes nécessitent des débits de communication et de performances de traitement toujours plus élevés tout en exigeant des durées de vie plus longues.

En considérant les contraintes et les exigences citées précédemment pour la conception des nœuds capteurs, différentes alternatives architecturales sont proposées et évaluées dans la littérature sous formes théoriques et expérimentales. Le microcontrôleur a été le processeur le plus utilisé dans la conception des nœuds de capteurs sans fil à cause de sa flexibilité, son coût et sa faible consommation énergétique. Etant donné que les capacités de traitement des nœuds capteurs à base des microcontrôleurs sont assez limitées, les circuits FPGA peuvent jouer un rôle clé dans les architectures des nœuds sous plusieurs formes : autonome, en combinaison avec les microcontrôleurs et des systèmes sur puce.

L'objectif principal de ce travail de thèse était la conception et la réalisation d'un nœud capteur sans fil à base d'une architecture embarquée sur un circuit FPGA et qui doit répondre à certaines exigences telles que la synchronisation de l'acquisition de données, un taux d'échantillonnage élevé, une transmission de données sécurisée et adapté pour les applications nécessitant un calcul intensif de données.

Dans ce travail, nous avons choisi le circuit FPGA Xilinx Artix-7 qui comprend une quantité suffisante de ressources logiques qui sont généralement requises dans les applications de surveillance de données. Les résultats montrent que l'implémentation sur le circuit donne une consommation de moins de 40% des ressources FPGA à l'exception des blocs BRAM (93%). Les résultats montrent aussi que plus de 75% d'énergie peut être conservée en commutant le nœud vers son état de veille.

La synchronisation temporelle est une fonctionnalité importante d'un RSCF qui influe directement sur la qualité et les caractéristiques de l'application. Pour assurer cette synchronisation de l'acquisition des nœuds capteurs, nous avons proposé un protocole de communication et de gestion entre les différents nœuds du réseau géré principalement par le nœud sink. Des tests pratiques ont été effectués dans deux environnements : l'un est favorable avec une visibilité directe, l'autre est défavorable en présence d'obstacles. Les résultats obtenus sont très satisfaisants, aussi bien en milieu idéal avec une erreur de synchronisation moyenne de 83,33 ns, qu'avec un milieu défavorable avec une erreur de 431,42 ns.

Afin d'effectuer une évaluation des nœuds développés dans un milieu réel, nous avons choisi une application de surveillance vibratoire des machines tournantes. La réalisation basée sur les RSCFs de ce type d'application, implique de grandes quantités de données à traiter et à transmettre sur l'ensemble du réseau. Comme il s'agit d'une application de surveillance industrielle, la sécurité est un facteur important, afin de sécuriser tout échange de données entre les différents nœuds du réseau. Plusieurs algorithmes ont été proposés dans la littérature, mais leur implémentation reste difficile à être appliquée, vu les contraintes des RSCFs, y compris une faible capacité de calcul, une faible mémoire, des ressources énergétiques limitées et l'utilisation de canaux de communication sans fil insécurisés. Pour cela, nous avons proposé une implémentation hardware de l'algorithme de cryptage AES 128 bits, sous forme d'un IP core, qui fait partie de l'architecture globale du nœud. L'architecture hardware permet une accélération dans le processus de cryptage et de décryptage, ce qui améliore les performances du nœud capteur. Les résultats montrent que le temps nécessaire pour le cryptage/

décryptage est de 505 ns pour une fréquence du signal d'horloge de 200 MHz, ou l'équivalent de 101 cycles, donc un débit de cryptage de 29,09 Mo/s, ce qui est très suffisant pour les applications des RCSFS.

En ce qui concerne le traitement embarqué pour une application de diagnostic de défauts dans les machines industrielles qui implique des traitements de calcul intensif tel que le calcul de la FFT, une architecture toujours hardware d'un bloc FFT a été réalisée. Dans cette architecture, un IP core FFT de Xilinx configuré en « Pipelined Streaming I/O » mode est utilisé, ce qui permet le traitement continu des données. Ce bloc FFT est testé avec plusieurs fréquences d'échantillonnage. Il a été trouvé, que des traitements de bonnes performances pour la surveillance des machines vibratoires à une fréquence d'échantillonnage de 10 kHz, peuvent être réalisés en utilisant des architectures basées sur un FPGA.

Dans le but de tester les nœuds développés et d'évaluer ses performances pour une application industrielle, plusieurs tests de diagnostic par analyse vibratoire pour la détection d'un défaut balourd est réalisé sur un banc d'essai. Pour ces tests, la faisabilité du traitement par le nœud capteur basé sur FPGA a été également évaluée pour deux fréquences d'échantillonnage, 400 Hz et 3,2 kHz. Les résultats montrent que, pour une fréquence d'échantillonnage de 3,2 kHz, le traitement par le nœud capteur proposé fournit non seulement les performances requises, mais aussi une consommation d'énergie acceptable. Dans ce cas, la durée de vie estimée en utilisant une batterie de 3,6 V et de capacité de 1000 mAh, est de plus de 47 heures pour un cycle de surveillance plus de 5 minutes.

Comme perspectives de ce travail, nous signalons que plus de 60 % des ressources du circuit FPGA ciblé, sont encore disponibles. Ces ressources peuvent être utilisées pour implémenter d'autres fonctionnalités au nœud capteur.

Dans ce contexte, nous envisageons, de renforcer la sécurité en ajoutant un mécanisme d'échange des clés dans le RCSF, par la conception d'un nouveau IP core qui implémente par exemple l'algorithme de cryptage par courbes elliptiques

dont l'intérêt particulier s'explique essentiellement par son niveau de sécurité très élevé et l'utilisation des clés de petites tailles.

En outre, d'autres techniques émergentes d'intelligence artificielle, y compris les réseaux de neurones, les systèmes experts et la logique floue, ont un grand intérêt à être utilisées dans les nœuds capteurs en tant que classificateurs de défauts.

BIBLIOGRAPHIE

- [1] J. Huang, G. Chen, L. Shu, Q. Zhang, and X. Wu, "WSNs-Based Mechanical Equipment State Monitoring and Fault Diagnosis in China," *Int. J. Distrib. Sens. Networks*, vol. 2015, no. 11, pp. 1–14, Nov. 2015.
- [2] L. Hou and N. W. Bergmann, "Novel industrial wireless sensor networks for machine condition monitoring and fault diagnosis," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 10, pp. 2787–2798, Oct. 2012.
- [3] Q. Huang, B. Tang, L. Deng, and J. Wang, "A divide-and-compress lossless compression scheme for bearing vibration signals in wireless sensor networks," *Measurement*, vol. 67, pp. 51–60, May 2015.
- [4] L. Hou and N. W. Bergmann, "System requirements for industrial wireless sensor networks," in *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, 2010, pp. 1–8.
- [5] Y. Yuan, Q. Xu, X. Guan, and Z. Liu, "Industrial high-speed wireless synchronous data acquisition system with real-time data compression," *Meas. J. Int. Meas. Confed.*, vol. 46, no. 9, pp. 3482–3487, Nov. 2013.
- [6] J. Huang, G. Chen, L. Shu, H. Lin, and K. Liu, "Designing Wireless Vibration Monitoring System for Petrochemical Units Fault Diagnosis," in *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing*, pp. 582–588.
- [7] B. Lu and V. C. Gungor, "Online and Remote Motor Energy Monitoring and Fault Diagnostics Using Wireless Sensor Networks," *IEEE Trans. Ind. Electron.*, vol. 56, no. 11, pp. 4651–4659, Nov. 2009.
- [8] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Networks*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008.
- [9] A. Moh'd, H. Marzi, N. Aslam, W. Phillips, and W. Robertson, "A secure platform of wireless sensor networks," *Procedia Comput. Sci.*, vol. 5, pp. 115–122, Jan. 2011.
- [10] P. Tang and T. W. S. Chow, "Wireless Sensor-Networks Conditions Monitoring and Fault Diagnosis Using Neighborhood Hidden Conditional Random Field," *IEEE Trans. Ind. Informatics*, vol. 12, no. 3, pp. 933–940, Jun. 2016.
- [11] J. Neuzil, O. Kreibich, and R. Smid, "A Distributed Fault Detection System Based on IWSN for Machine Condition Monitoring," *IEEE Trans. Ind. Informatics*, vol. 10, no. 2, pp. 1118–1123, May 2014.
- [12] V. Gungor and G. Hancke, "Industrial Wireless Sensor Networks: Applications, Protocols, and Standards," Apr-2013. [Online].
- [13] Z. Wang and C. S. Chang, "Online fault detection of induction motors using frequency domain independent components analysis," in *2011 IEEE International Symposium on Industrial Electronics*, 2011, pp. 2132–2137.
- [14] L. M. Contreras-Medina, R. D. J. Romero-Troncoso, E. Cabal-Yepez, J. D. J.

- Rangel-Magdaleno, and J. R. Millan-Almaraz, "FPGA-based multiple-channel vibration analyzer for industrial applications in induction motor failure detection," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 1, pp. 63–72, Jan. 2010.
- [15] C. Buccella, C. Cecati, and H. Latafat, "Digital control of power converters - A survey," *IEEE Trans. Ind. Informatics*, vol. 8, no. 3, pp. 437–447, Aug. 2012.
- [16] E. Monmasson, L. Idkhajine, M. N. Cirstea, I. Bahri, A. Tisan, and M. W. Naouar, "FPGAs in Industrial Control Applications," *IEEE Trans. Ind. Informatics*, vol. 7, no. 2, pp. 224–243, May 2011.
- [17] A. de la Piedra, A. Braeken, and A. Touhafi, "Sensor Systems Based on FPGAs and Their Applications: A Survey," *Sensors*, vol. 12, no. 9, pp. 12235–12264, 2012.
- [18] A. Araujo, J. Garcia-Palacios, J. Blesa, F. Tirado, E. Romero, A. Samartin, and O. Nieto-Taladriz, "Wireless Measurement System for Structural Health Monitoring With High Time-Synchronization Accuracy," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 3, pp. 801–810, 2012.
- [19] Q. Huang, B. Tang, and L. Deng, "Development of high synchronous acquisition accuracy wireless sensor network for machine vibration monitoring," *Meas. J. Int. Meas. Confed.*, vol. 66, pp. 35–44, Apr. 2015.
- [20] H. A. Salam and B. M. Khan, "IWSN - Standards, Challenges and Future," *IEEE Potentials*, vol. 35, no. 2, pp. 9–16, Mar. 2016.
- [21] V. C. Gungor and G. P. Hancke, "Industrial Wireless Sensor Networks: Challenges, Design Principles, and Technical Approaches," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 4258–4265, 2009.
- [22] S. Raza, A. Slabbert, T. Voigt, and K. Landernas, "Security considerations for the WirelessHART protocol," in *2009 IEEE Conference on Emerging Technologies & Factory Automation*, 2009, pp. 1–8.
- [23] K. Al Agha, M. H. Bertin, T. Dang, A. Guitton, P. Minet, T. Val, and J. B. Viollet, "Which Wireless Technology for Industrial Wireless Sensor Networks? The Development of OCARI Technology," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 4266–4278, Oct. 2009.
- [24] R. Falk and H.-J. Hof, "Industrial Sensor Network Security Architecture," in *2010 Fourth International Conference on Emerging Security Information, Systems and Technologies*, 2010, pp. 97–102.
- [25] K. Islam, W. Shen, and X. Wang, "Wireless Sensor Network Reliability and Security in Factory Automation: A Survey," *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.)*, vol. 42, no. 6, pp. 1243–1256, Nov. 2012.
- [26] Chee-Yee Chong and S. P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," *Proc. IEEE*, vol. 91, no. 8, pp. 1247–1256, Aug. 2003.
- [27] A. H. M. Jallad, "Distributed computing in space-based wireless sensor networks," University of Surrey (United Kingdom), 2008.
- [28] O. Yang, "Sleeping strategies for wireless sensor networks," University of Rochester, 2011.
- [29] Y. M. Liu, S. C. Wu, and X. H. Nian, "The architecture and characteristics of

- wireless sensor network,” *ICCTD 2009 - 2009 Int. Conf. Comput. Technol. Dev.*, vol. 1, no. 561, pp. 561–565, 2009.
- [30] H. M. A. Fahmy, *Wireless sensor networks: concepts, applications, experimentation and analysis*. Springer, 2016.
- [31] Muhammad R Ahmed and Xu Huang and Dharmendra Sharma and Hongyan Cui, “Wireless Sensor Network: Characteristics and Architectures,” *Int. J. Electr. Comput. Energ. Electron. Commun. Eng. Vol6*, vol. 6, no. 12, pp. 1398–1401, 2012.
- [32] J. a. Stankovic, “Research challenges for wireless sensor networks,” *ACM SIGBED Rev.*, vol. 1, no. 2, pp. 9–12, 2004.
- [33] W. Dargie and C. Poellabauer, *Fundamentals of wireless sensor networks: theory and practice*. John Wiley & Sons, 2010.
- [34] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, “Wireless sensor networks: a survey on recent developments and potential synergies,” *J. Supercomput.*, vol. 68, no. 1, pp. 1–48, 2014.
- [35] D. Chen and P. K. Varshney, “QoS Support in Wireless Sensor Networks: A Survey.” in *International conference on wireless networks*, 2004, vol. 233, pp. 1–7.
- [36] H. M. Ammari, *The Art of Wireless Sensor Networks*, vol. 1. 2014.
- [37] T. Arampatzis, J. Lygeros, and S. Manesis, “A Survey of Applications of Wireless Sensors and Wireless Sensor Networks,” in *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.*, 2005, pp. 719–724.
- [38] A. Perrig, J. Stankovic, and D. Wagner, “Security in wireless sensor networks,” *Commun. ACM*, vol. 47, no. 6, pp. 53–57, 2004.
- [39] C. Buratti, A. Conti, D. Dardari, and R. Verdone, “An overview on wireless sensor networks technology and evolution,” *Sensors*, vol. 9, no. 9, pp. 6869–6896, 2009.
- [40] D. Anouar, “Contributions à l’amélioration des performances des réseaux de capteurs sans fil à base d’IR-UWB,” Université Mohammed V-Agdal, Faculté des Sciences, Rabat, 2015.
- [41] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, “A taxonomy of wireless micro-sensor network models,” *ACM SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 6, no. 2, pp. 28–36, Apr. 2002.
- [42] Y. L. M. T. T. W. Wu, *Wireless Sensor Networks and Applications*. Springer Berlin Heidelberg, 2008.
- [43] A. Karl, Holger and Willig, *Protocols and Architectures for Wireless Sensor Networks*. Chichester, UK: John Wiley & Sons, Ltd, 2005.
- [44] K. Shahzad, “Energy Efficient Wireless Sensor Node Architecture for Data and Computation Intensive Applications,” Mid Sweden University, 2014.
- [45] M. J. McGrath and C. N. Scanail, *Sensor Technologies: Healthcare, Wellness and Environmental Applications*. Apress, 2013.
- [46] F. Karray, M. W. Jmal, M. Abid, M. S. Bensaleh, and A. M. Obeid, “A review

- on wireless sensor node architectures,” in *2014 9th International Symposium on Reconfigurable and Communication-Centric Systems-on-Chip, ReCoSoC 2014*, 2014, pp. 1–8.
- [47] A. Sabato, C. Niezrecki, and G. Fortino, “Wireless MEMS-Based Accelerometer Sensor Boards for Structural Vibration Monitoring: A Review,” *IEEE Sens. J.*, vol. 17, no. 2, pp. 226–235, 2017.
- [48] S. Sohn, “Feasibility study on the use of wireless accelerometers in the experimental modal testing,” *J. Supercomput.*, vol. 72, no. 7, 2016.
- [49] S. Y. Chiang, Y. C. Kan, Y. S. Chen, Y. C. Tu, and H. C. Lin, “Fuzzy computing model of activity recognition on WSN movement data for ubiquitous healthcare measurement,” *Sensors (Switzerland)*, vol. 16, no. 12, pp. 1–22, 2016.
- [50] A. Somov, Z. J. Chew, T. Ruan, Q. Li, and M. Zhu, “Poster Abstract: Piezoelectric Energy Harvesting Powered WSN for Aircraft Structural Health Monitoring,” *2016 15th ACM/IEEE Int. Conf. Inf. Process. Sens. Networks, IPSN 2016 - Proc.*, pp. 4–5, 2016.
- [51] N. Margiotta, G. Avitabile, and G. Coviello, “A wearable wireless system for gait analysis for early diagnosis of Alzheimer and Parkinson disease,” in *2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA)*, 2016, pp. 1–4.
- [52] Puvaneshwari S and Vijayashaarathi S, “Efficient Monitoring system for cardiac patients using Wireless Sensor Networks (WSN),” in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2016, pp. 1558–1561.
- [53] X. Bao, H. Li, D. Xu, L. Jia, B. Ran, and J. Rong, “Traffic vehicle counting in jam flow conditions using low-cost and energy-efficient wireless magnetic sensors,” *Sensors (Switzerland)*, vol. 16, no. 11, 2016.
- [54] H. Zhu and F. Yu, “A Cross-Correlation Technique for Vehicle Detections in Wireless Magnetic Sensor Network,” *IEEE Sens. J.*, vol. 16, no. 11, pp. 4484–4494, 2016.
- [55] V. Mandalapa Bhoopathy, M. Ben Haj Frej, S. Richard Ebenezer Amalorpavaraj, and I. Shaik, “Localization and mobility of underwater acoustic sensor nodes,” in *2016 Annual Connecticut Conference on Industrial Electronics, Technology & Automation (CT-IETA)*, 2016, pp. 1–5.
- [56] K. Wang, H. Gao, X. Xu, J. Jiang, and D. Yue, “An Energy-Efficient Reliable Data Transmission Scheme for Complex Environmental Monitoring in Underwater Acoustic Sensor Networks,” *IEEE Sens. J.*, vol. 16, no. 11, pp. 4051–4062, 2016.
- [57] A. Arfire, A. Marjovi, and A. Martinoli, “Mitigating Slow Dynamics of Low-Cost Chemical Sensors for Mobile Air Quality Monitoring Sensor Networks,” *Int. Conf. Embed. Wirel. Syst. Networks*, pp. 1159–167, 2016.
- [58] I. Stoianov, L. Nachman, S. Madden, and T. Tokmouline, “PIPENETa wireless sensor network for pipeline monitoring,” *Proc. 6th Int. Conf. Inf. Process. Sens. networks - IPSN '07*, p. 264, 2007.
- [59] Y. Zhao, J. R. Smith, and A. Sample, “NFC-WISP: A sensing and

- computationally enhanced near-field RFID platform,” in *2015 IEEE International Conference on RFID (RFID)*, 2015, vol. 2, no. 9, pp. 174–181.
- [60] I. Downes, L. B. Rad, and H. Aghajan, “Development of a mote for wireless image sensor networks,” *Proc. Cogn. Syst. with Interact. Sensors (COGIS)*, Paris, Fr., no. March, p. 8, 2006.
- [61] K. Yifan and J. Peng, “Development of Data Video Base Station in Water Environment Monitoring Oriented Wireless Sensor Networks,” in *2008 International Conference on Embedded Software and Systems Symposia*, 2008, pp. 281–286.
- [62] Q. Lu, L. Du, and B. Hu, “Low-Power JPEG2000 Implementation on DSP-Based Camera Node in Wireless Multimedia Sensor Networks,” in *2009 International Conference on Networks Security, Wireless Communications and Trusted Computing*, 2009, vol. 1, pp. 300–303.
- [63] H. Asada, G and Dong, M and Lin, TS and Newberg, F and Pottie, G and Kaiser, WJ and Marcy, “Wireless Integrated Sensors Network: Low power systems on a chip,” in *European Solid State Circuits Conference*, 1998.
- [64] Xiaoyu Zhang, Hanjun Jiang, Lingwei Zhang, Chun Zhang, Zhihua Wang, and Xinkai Chen, “An Energy-Efficient ASIC for Wireless Body Sensor Networks in Medical Applications,” *IEEE Trans. Biomed. Circuits Syst.*, vol. 4, no. 1, pp. 11–18, 2010.
- [65] H. Bhamra, J. Lynch, M. Ward, and P. Irazoqui, “A Noise-Power-Area Optimized Biosensing Front End for Wireless Body Sensor Nodes and Medical Implantable Devices,” *IEEE Trans. Very Large Scale Integr. Syst.*, pp. 1–12, 2017.
- [66] M. Kohvakka, T. Arpinen, M. Hännikäinen, and T. D. Hämäläinen, “High-performance multi-radio WSN platform,” in *Proceedings of the second international workshop on Multi-hop ad hoc networks: from theory to reality - REALMAN '06*, 2006, p. 95.
- [67] R. M. Voyles, S. Povilus, R. Mangharam, and K. Li, “RecoNode: A reconfigurable node for heterogeneous multi-robot search and rescue,” in *2010 IEEE Safety Security and Rescue Robotics*, 2010, vol. i, pp. 1–7.
- [68] T. Nyländen, J. Boutellier, K. Nikunen, J. Hannuksela, and O. Silvén, “Low-Power Reconfigurable Miniature Sensor Nodes for Condition Monitoring,” *Int. J. Parallel Program.*, vol. 43, no. 1, pp. 3–23, Jan. 2014.
- [69] F. Philipp, “Runtime Hardware Reconfiguration in Wireless Sensor Networks for Condition Monitoring,” *Universitäts- und Landesbibliothek Darmstadt*, 2014.
- [70] W.-J. Yi, S. Gilliland, and J. Saniie, “Wireless sensor network for structural health monitoring using System-on-Chip with Android smartphone,” *2013 IEEE Sensors*, pp. 1–4, 2013.
- [71] M. Imran, K. Shahzad, N. Ahmad, M. O’Nils, N. Lawal, and B. Oelmann, “Energy-Efficient SRAM FPGA-Based Wireless Vision Sensor Node: SENTIOF-CAM,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 12, pp. 2132–2143, Dec. 2014.
- [72] O. Berder and O. Sentieys, “PowWow : Power Optimized Hardware/Software

- Framework for Wireless Motes,” in *Proc. 23rd International Conference on Architecture of Computing Systems (ARCS)*, 2010, pp. 1–5.
- [73] Y. Hong-fang, W. Hua-qing, Q. He, and K. Xi-yong, “A ZigBee wireless sensor network for fault diagnosis of pump,” in *Natural Computation (ICNC), 2010 Sixth International Conference on*, 2010, vol. 7, pp. 3778–3782.
- [74] R. Faludi, *A Practical Guide to the ZigBee Mesh Networking Protocol. Building Wireless Sensor Networks*, no. 1. 2010.
- [75] H. Liptak, B and Eren, *Instrument Engineers’ Handbook, Volume 3: Process Software and Digital Networks*, vol. III. CRC Press, 2012.
- [76] J. Lee, Y. Su, and C. Shen, “A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi,” in *IECON 2007 - 33rd Annual Conference of the IEEE Industrial Electronics Society*, 2007, pp. 46–51.
- [77] Y. Idris and N. A. Muhammad, “A Comparative Study of Wireless Communication Protocols: Zigbee vs Bluetooth,” *Int. J. Eng. Sci. Comput.*, vol. 6, no. 4, pp. 3741–3744, 2016.
- [78] A. N. Kim, F. Hekland, S. Petersen, and P. Doyle, “When HART goes wireless: Understanding and implementing the WirelessHART standard,” in *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, 2008, pp. 899–907.
- [79] A. and others Kabir, Md and Khan, Md and Ali, “COMPARISON AMONG SHORT RANGE WIRELESS NETWORKS: BLUETOOTH, ZIGBEE, & WI-FI,” *DAFFODIL Int. Univ. J. Sci. Technol.*, vol. 11, no. 1, 2016.
- [80] F. Vahid and T. Givargis, *Embedded system design: a unified hardware/software introduction*. New York: Wiley, 2002.
- [81] R. Kamal, *Embedded systems: architecture, programming and design*. Tata McGraw-Hill Education, 2011.
- [82] T. D. Morton, *Embedded Microcontrollers*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000.
- [83] S. Heath, *Embedded systems design*. Newnes, 2002.
- [84] M. I. B. Khelifa, “contribution of FPGA-based System-on-Chip controllers for embedded AC drive electrical applications,” Université de Cergy Pontoise, 2011.
- [85] A. Nohl, F. Schirrmeister, and D. Taussig, “Application specific processor design: Architectures, design methods and tools,” in *2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2010, pp. 349–352.
- [86] C. Wilhelmsson, *Embedded Systems and FPGAs for Implementation of Control Oriented Models, Applied to Combustion Engines*. Lund University, 2009.
- [87] G. Kappen, L. Kurz, and T. G. Noll, “Comparison of ASIP and Standard Microprocessor based Navigation Processors,” *Chair Electr. Eng. Comput. Syst. Aachen Univ. Ger.*, 2007.
- [88] E. Monmasson and M. N. Cirstea, “FPGA design methodology for industrial

- control systems—A review,” *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1824–1842, 2007.
- [89] S. Brown and J. Rose, “Architecture of FPGAs and CPLDs: A tutorial,” *IEEE Des. Test Comput.*, vol. 13, no. 2, pp. 42–57, 1996.
- [90] N. Alexandre, *Circuits logiques programmables - Mémoires PLD, CPLD et FPGA*, Ellipses. Ellipses, 1998.
- [91] C. Maxfield, *The Design Warrior’s Guide to FPGAs: Devices, Tools and Flows*, 1st ed. Newton, MA, USA: Newnes, 2004.
- [92] R. Dubey, P. Agarwal, and M. K. Vasantha, “Programmable Logic Devices for Motion Control - A Review,” *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 559–566, Feb. 2007.
- [93] U. Farooq, Z. Marrakchi, and H. Mehrez, “FPGA Architectures: An Overview,” in *Tree-based Heterogeneous FPGA Architectures*, New York, NY: Springer New York, 2012, pp. 7–48.
- [94] D. Buell, T. El-Ghazawi, K. Gaj, and V. Kindratenko, “High-performance reconfigurable computing,” *Comput. Comput. Soc.*, vol. 40, no. 3, p. 23, 2007.
- [95] H. Yang, J. Zhang, J. Sun, and L. Yu, “Review of advanced FPGA architectures and technologies,” *J. Electron.*, vol. 31, no. 5, pp. 371–393, Oct. 2014.
- [96] M. Niknahad, *Using Fine Grain Approaches for Highly Reliable Design of FPGA-based Systems in Space*. KIT Scientific Publishing, 2013.
- [97] N. A. Wang, Laung-Terng and Stroud, Charles E and Touba, *System-on-chip test architectures: nanometer design for testability*. Morgan Kaufmann, 2010.
- [98] R. Rajsuman, *System-on-a-chip: Design and Test*. Artech House, Inc., 2000.
- [99] K. MESSAOUDI, “Traitement des Signaux et Images en Temps Réel « Implantation de H.264 sur MPSoC »,” Université Badji Mokhtar, 2012.
- [100] E. Yasuura, Hiroto and Tomiyama, Hiroyuki and Inoue, Akihiko and Fajar, “Embedded System Design Using Soft-Core Processor and Valen-C,” *J. Inf. Sci. Eng.*, vol. 14, pp. 587–603, 1998.
- [101] J. G. Tong, I. D. L. Anderson, and M. A. S. Khalid, “Soft-Core Processors for Embedded Systems,” in *2006 International Conference on Microelectronics*, 2006, pp. 170–173.
- [102] W. Wolf, “A decade of hardware/ software codesign,” *Computer (Long. Beach. Calif.)*, vol. 36, no. 4, pp. 38–43, Apr. 2003.
- [103] G. De Michell and R. K. Gupta, “Hardware/software co-design,” *Proc. IEEE*, vol. 85, no. 3, pp. 349–365, Mar. 1997.
- [104] S. Scalera, M. Falco, and B. Nelson, “A reconfigurable computing architecture for microsensors,” in *Proceedings 2000 IEEE Symposium on Field-Programmable Custom Computing Machines (Cat. No.PR00871)*, 2000, pp. 59–67.
- [105] J. M. Rabaey, M. J. Ammer, J. L. da Silva, D. Patel, and S. Roundy, “PicoRadio supports ad hoc ultra-low power wireless networking,” *Computer (Long. Beach. Calif.)*, vol. 33, no. 7, pp. 42–48, Jul. 2000.

- [106] M. A. M. Vieira, C. N. Coelho, D. C. Da Silva, and J. M. Da Mata, "Survey on wireless sensor network devices," in *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, 2003, vol. 1, no. January, pp. 537–544.
- [107] S. Z. Ahmed, G. Sassatelli, L. Torres, and L. Rouge, "Survey of New Trends in Industry for Programmable Hardware: FPGAs, MPPAs, MPSoCs, Structured ASICs, eFPGAs and New Wave of Innovation in FPGAs," in *2010 International Conference on Field Programmable Logic and Applications*, 2010, vol. 1, pp. 291–297.
- [108] A. de la Piedra, A. Braeken, and A. Touhafi, "Sensor systems based on FPGAs and their applications: A survey," *Sensors (Switzerland)*, vol. 12, no. 9, pp. 12235–12264, Sep. 2012.
- [109] M. L. Kaddachi, A. Soudani, I. Noura, V. Lecuire, and K. Torki, "Efficient hardware solution for low power and adaptive image-compression in WSN," in *2010 17th IEEE International Conference on Electronics, Circuits and Systems*, 2010, no. 1, pp. 583–586.
- [110] A. Chefi, A. Soudani, and G. Sicard, "Hardware compression solution based on HWT for low power image transmission in WSN," in *ICM 2011 Proceeding*, 2011, vol. 2000, pp. 1–5.
- [111] L. a. Vera-Salas, S. V. Moreno-Tapia, R. a. Osornio-Rios, and R. de_J. Romero-Troncoso, "Reconfigurable Node Processing Unit for a Low-Power Wireless Sensor Network," in *2010 International Conference on Reconfigurable Computing and FPGAs*, 2010, pp. 173–178.
- [112] L. Gasparini, R. Manduchi, M. Gottardi, and D. Petri, "An Ultralow-Power Wireless Camera Node: Development and Performance Analysis," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 12, pp. 3824–3832, Dec. 2011.
- [113] A. Ledeczki, T. Hay, P. Volgyesi, D. R. Hay, A. Nadas, and S. Jayaraman, "Wireless Acoustic Emission Sensor Network for Structural Monitoring," *IEEE Sens. J.*, vol. 9, no. 11, pp. 1370–1377, Nov. 2009.
- [114] R. Garcia, A. Gordon-Ross, and A. D. George, "Exploiting Partially Reconfigurable FPGAs for Situation-Based Reconfiguration in Wireless Sensor Networks," in *2009 17th IEEE Symposium on Field Programmable Custom Computing Machines*, 2009, pp. 243–246.
- [115] P. Turcza and M. Duplaga, "Low power FPGA-based image processing core for wireless capsule endoscopy," *Sensors Actuators, A Phys.*, vol. 172, no. 2, pp. 552–560, 2011.
- [116] D. M. Pham and S. M. Aziz, "FPGA architecture for object extraction in Wireless Multimedia Sensor Network," in *2011 Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2011, pp. 294–299.
- [117] F. Philipp, F. Samman, and M. Glesner, "Real-time Characterization of Noise Sources with Computationally Optimised Wireless Sensor Networks Fran , Localization of acoustic sources," in *DAGA Jahrestagung*, 2011, pp. 713–714.
- [118] F. Philipp, F. A. Samman, and M. Glesner, "Design of an autonomous platform for distributed sensing-actuating systems," in *2011 22nd IEEE International*

Symposium on Rapid System Prototyping, 2011, pp. 85–90.

- [119] J. Liao, B. K. Singh, M. A. Khalid, and K. E. Tepe, "FPGA based wireless sensor node with customizable event-driven architecture," *EURASIP J. Embed. Syst.*, vol. 2013, no. 1, p. 5, Dec. 2013.
- [120] A. El Kateeb, A. Ramesh, and L. Azzawi, "Wireless Sensor Nodes Processor Architecture and Design," in *22nd International Conference on Advanced Information Networking and Applications - Workshops (aina workshops 2008)*, 2008, pp. 892–897.
- [121] A. Elkateeb, "RH-mote for Next-generation Wireless Sensor Networks," *Procedia Comput. Sci.*, vol. 21, pp. 217–224, 2013.
- [122] Y. Sun, L. Li, and H. Luo, "Design of FPGA-based multimedia node for WSN," in *7th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2011*, 2011, pp. 1–5.
- [123] S. Tanaka, N. Fujita, Y. Yanagisawa, T. Terada, and M. Tsukamoto, "Reconfigurable hardware architecture for saving power consumption on a sensor node," in *2008 International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2008, pp. 405–410.
- [124] J. Rabaey and A. Way, "PICORADIO : COMMUNICATION / COMPUTATION PICONODES FOR SENSOR NETWORKS Space Vehicles Directorate 3550 Aberdeen Ave SE AIR FORCE MATERIEL COMMAND," Berkeley Wireless Research Center 2108 Allston Way, Suite 200 Berkeley, CA 94704-1698, 2003.
- [125] J. Portilla, A. De Castro, and T. Riesgo, "A Modular Architecture for Nodes in Wireless Sensor Networks," *J. Univers. Comput. Sci.*, vol. 12, no. 3, pp. 328–339, 2006.
- [126] J. Portilla, T. Riesgo, and A. de Castro, "A Reconfigurable Fpga-Based Architecture for Modular Nodes in Wireless Sensor Networks," in *2007 3rd Southern Conference on Programmable Logic*, 2007, pp. 203–206.
- [127] Y. E. Krasteva, J. Portilla, J. M. Carnicer, E. De La Torre, and T. Riesgo, "Remote HW-SW reconfigurable wireless sensor nodes," in *Proceedings - 34th Annual Conference of the IEEE Industrial Electronics Society, IECON 2008*, 2008, pp. 2483–2488.
- [128] Y. E. Krasteva, J. Portilla, E. De La Torre, and T. Riesgo, "Embedded runtime reconfigurable nodes for wireless sensor networks applications," *IEEE Sens. J.*, vol. 11, no. 9, pp. 1800–1810, Sep. 2011.
- [129] G. Simon, M. Maróti, Á. Lédeczi, G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai, and K. Frampton, "Sensor network-based countersniper system," in *Proceedings of the 2nd international conference on Embedded networked sensor systems - SenSys '04*, 2004, p. 1.
- [130] S. J. Bellis, K. Delaney, B. O'Flynn, J. Barton, K. M. Razeeb, and C. O'Mathuna, "Development of field programmable modular wireless sensor network nodes for ambient systems," *Comput. Commun.*, vol. 28, no. 13 SPEC. ISS., pp. 1531–1544, 2005.
- [131] B. O'Flynn, S. Bellis, K. Delaney, J. Barton, S. C. O'Mathuna, A. M. Barroso, J. Benson, U. Roedig, and C. Sreenan, "The development of a novel

- minaturized modular platform for wireless sensor networks,” in *IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005.*, 2005, vol. 2005, pp. 370–375.
- [132] R. B. Caldas, F. L. Correa, J. Augusto Nacif, T. R. Roque, L. B. Ruiz, A. O. Fernandes, J. M. da Mata, and C. Coelho, “Low Power/High Performance Self-Adapting Sensor Node Architecture,” in *2005 IEEE Conference on Emerging Technologies and Factory Automation*, 2005, vol. 2, pp. 973–976.
- [133] K. Virk, J. Madsen, A. V. Lorentzen, M. Leopold, and P. Bonnet, “Design of A Development Platform for HW/SW Codesign of Wireless Integrated Sensor Nodes,” in *8th Euromicro Conference on Digital System Design (DSD’05)*, 2005, pp. 254–260.
- [134] J. Valverde, A. Otero, M. Lopez, J. Portilla, E. de la Torre, and T. Riesgo, “Using SRAM based FPGAs for power-aware high performance wireless sensor networks,” *Sensors*, vol. 12, no. 3, pp. 2667–2692, 2012.
- [135] J. Koskinen, P. Kilpeläinen, J. Rehu, P. Tukeva, and M. Sallinen, “Wireless sensor networks for infrastructure and industrial monitoring applications,” *2010 Int. Conf. Inf. Commun. Technol. Converg. ICTC 2010*, pp. 250–255, 2010.
- [136] C.-M. Hsieh, Z. Wang, and J. Henkel, “A Reconfigurable Hardware Accelerated Platform for Clustered Wireless Sensor Networks,” in *2012 IEEE 18th International Conference on Parallel and Distributed Systems*, 2012, pp. 498–505.
- [137] K. Shahzad, P. Cheng, and B. Oelmann, “Architecture exploration for a high-performance and low-power wireless vibration analyzer,” *IEEE Sens. J.*, vol. 13, no. 2, pp. 670–682, 2013.
- [138] K. Shahzad and B. Oelmann, “An FPGA-based high-performance wireless vibration analyzer,” in *2013 NORCHIP*, 2013, pp. 1–5.
- [139] S. Commuri, V. Tadigotla, and M. Atiquzzaman, “Reconfigurable Hardware Based Dynamic Data Aggregation in Wireless Sensor Networks,” *Int. J. Distrib. Sens. Networks*, vol. 4, no. 2, pp. 194–212, May 2008.
- [140] J. Tong, Z. Zhang, Q. Sun, and Z. Chen, “Design of Wireless Sensor Network Node with Hyperchaos Encryption Based on FPGA,” in *International Workshop on Chaos-Fractals Theories and Applications Design*, 2009, pp. 190–194.
- [141] Y. Li, Z. Jia, F. Liu, and S. Xie, “Hardware reconfigurable wireless sensor network node with power and area efficiency,” *IET Wirel. Sens. Syst.*, vol. 2, no. 3, p. 247, 2012.
- [142] C. Hu, Z. Liu, Y. Pan, Z. Zeng, and M. Q. H. Meng, “A novel FPGA-based wireless vision sensor node,” in *Proceedings of the 2009 IEEE International Conference on Automation and Logistics, ICAL 2009*, 2009, pp. 841–846.
- [143] C. Plessl, R. Enzler, H. Walder, J. Beutel, M. Platzner, L. Thiele, and G. Troster, “The case for reconfigurable hardware in wearable computing,” *Pers. Ubiquitous Comput.*, vol. 7, no. 5, pp. 299–308, Oct. 2003.
- [144] P. Volgyesi, G. Balogh, A. Nadas, C. B. Nash, and A. Ledeczi, “Shooter

- Localization and Weapon Classification with Soldier-Wearable Networked Sensors,” in *Proceedings of the 5th international conference on Mobile systems, applications and services*, 2007, pp. 113–126.
- [145] T. Ahola, P. Korpinen, J. Rakkola, and R. Teemu, “Wearable FPGA Based Wireless Sensor Platform,” in *Proceedings of the 29th Annual International Conference of the IEEE EMBS*, 2007, pp. 2288–2291.
- [146] H. Hinkelmann, A. Reinhardt, S. Varyani, and M. Glesner, “A Reconfigurable Prototyping Platform for Smart Sensor Networks,” in *2008 4th Southern Conference on Programmable Logic*, 2008, pp. 125–130.
- [147] H. Hinkelmann, A. Reinhardt, and M. Glesner, “A Methodology for Wireless Sensor Network Prototyping with Sophisticated Debugging Support,” in *2008 The 19th IEEE/IFIP International Symposium on Rapid System Prototyping*, 2008, pp. 82–88.
- [148] H. Hinkelmann, P. Zipf, and M. Glesner, “Design and evaluation of an energy-efficient dynamically reconfigurable architecture for wireless sensor nodes,” in *2009 International Conference on Field Programmable Logic and Applications*, 2009, pp. 359–366.
- [149] G. Chalivendra, R. Srinivasan, and N. S. Murthy, “FPGA based reconfigurable wireless sensor network protocol,” in *2008 International Conference on Electronic Design*, 2008, pp. 1–4.
- [150] P. Muralidhar and C. B. R. Rao, “Reconfigurable Wireless sensor network node based on NIOS core,” in *Proceedings of the 4th International Conference on Wireless Communication and Sensor Networks, WCSN 2008*, 2008, pp. 67–72.
- [151] S. Lu, X. Huang, L. Cui, Z. Zhao, and D. Li, “Design and implementation of an ASIC-based sensor device for WSN applications,” *IEEE Trans. Consum. Electron.*, vol. 55, no. 4, pp. 1959–1967, 2009.
- [152] R. S. Allgayer, M. Götz, and C. E. Pereira, “FemtoNode: Reconfigurable and customizable architecture for wireless sensor networks,” in *IFIP Advances in Information and Communication Technology*, vol. 310, A. Rettberg, M. C. Zanella, M. Amann, M. Keckeisen, and F. J. Rammig, Eds. Springer Berlin Heidelberg, 2009, pp. 302–309.
- [153] Jiasheng Wei, Ling Wang, Feng Wu, Yibo Chen, and Long Ju, “Design and implementation of wireless sensor node based on open core,” in *2009 IEEE Youth Conference on Information, Computing and Telecommunication*, 2009, pp. 102–105.
- [154] Hongzhi Liu and N. W. Bergmann, “An FPGA softcore based implementation of a bird call recognition system for sensor networks,” in *2010 Conference on Design and Architectures for Signal and Image Processing (DASIP)*, 2010, pp. 1–6.
- [155] A. Sumarudin, T. Adiono, and W. P. Putra, “Flexible and reconfigurable system on chip for wireless sensor network,” *2014 Int. Conf. Inf. Technol. Syst. Innov. ICITSI 2014 - Proc.*, no. November, pp. 230–234, 2014.
- [156] A. Gaga, O. Diouri, and F. Errahimi, “Design and Implementation of Wireless Zigbee Sensor Based On Embedded 32-Bits FPGA Processor,” *Mediterr.*

- Telecommun. J.*, vol. 5, no. June, pp. 178–183, 2015.
- [157] X. Zhai and T. Vladimirova, “Data Aggregation in Wireless Sensor Networks for Lunar Exploration,” in *2015 Sixth International Conference on Emerging Security Technologies (EST)*, 2015, no. 312826, pp. 30–37.
- [158] R. Keerthana, K and Prashanthi, “Flexible and Reconfigurable SOC for Sensor Network under ZYNQ Processor,” *Int. J. Sci. Technol. Eng.*, vol. 3, no. 1, pp. 145–151, 2016.
- [159] Z. Zhou and L. Zhang, “Large Scale Satellite-Based Wireless Sensor Networks for Arctic Monitoring,” in *Arctic Technology Conference*, 2016.
- [160] B. Bengherbia, S. Chadli, M. O. Zmirli, and A. Toubal, “A MicroBlaze based WSN sink node using XBee transceiver,” in *Proceedings of 2016 8th International Conference on Modelling, Identification and Control, ICMIC 2016*, 2017.
- [161] C. Paar and J. Pelzl, *Understanding cryptography: a textbook for students and practitioners*. Springer Science & Business Media, 2009.
- [162] P. Hamalainen, M. Hannikainen, and T. D. Hamalainen, “Review of Hardware Architectures for Advanced Encryption Standard Implementations Considering Wireless Sensor Networks,” in *Proceedings of the 6th International Workshop on Systems, Architectures, Modeling, and Simulation*, S. Vassiliadis, M. Bereković, and T. D. Hämäläinen, Eds. Springer Berlin Heidelberg, 2006, pp. 443–453.
- [163] L. A. Cardona, B. Lorente, and C. Ferrer, “Partial crypto-reconfiguration of nodes based on FPGA for WSN,” in *Proceedings - International Carnahan Conference on Security Technology*, 2014, vol. 2014–Octob, no. October, pp. 1–4.
- [164] S. Drimer, T. Güneysu, and C. Paar, “DSPs, BRAMs and a pinch of logic: New recipes for AES on FPGAs,” in *Proceedings of the 16th IEEE Symposium on Field-Programmable Custom Computing Machines, FCCM’08*, 2008, pp. 99–108.
- [165] H. Houssain and M. Badra, “Hardware Implementations of Elliptic Curve Cryptography in Wireless Sensor Networks,” 2011, no. December, pp. 11–14.
- [166] O. Kreibich, J. Neuzil, and R. Smid, “Quality-based multiple-sensor fusion in an industrial wireless sensor network for MCM,” *IEEE Trans. Ind. Electron.*, vol. 61, no. 9, pp. 4903–4911, Sep. 2014.
- [167] a. Albarbar, S. Mekid, a G. Starr, and R. Pietruszkiewicz, “Suitability of MEMS accelerometers for condition monitoring: An experimental study,” *Sensors*, vol. 8, no. 2, pp. 784–799, Feb. 2008.
- [168] I. S. Bozchalooi and M. Liang, “Parameter-free bearing fault detection based on maximum likelihood estimation and differentiation,” *Meas. Sci. Technol.*, vol. 20, no. 6, p. 65102, Jun. 2009.
- [169] F. Li, L. Ye, G. C. Zhang, and G. Meng, “Bearing Fault Detection Using Higher-Order Statistics Based ARMA Model,” *Key Eng. Mater.*, vol. 347, pp. 271–276, 2007.
- [170] F. Philipp, J. Martinez, M. Glesner, and A. Arkkio, “A smart wireless sensor

- for the diagnosis of broken bars in induction motors,” in *2012 13th Biennial Baltic Electronics Conference*, 2012, no. 1, pp. 119–122.
- [171] J. Martinez, F. Philipp, M. Glesner, and A. Arkkio, “An accurate and fast technique for correcting spectral leakage in motor diagnosis,” in *2013 9th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives (SDEMPED)*, 2013, no. 3, pp. 215–220.
- [172] S. Ganeriwal, R. Kumar, and M. B. Srivastava, “Timing-sync protocol for sensor networks,” in *Proceedings of the first international conference on Embedded networked sensor systems - SenSys '03*, 2003, p. 138.
- [173] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, “The flooding time synchronization protocol,” in *Proceedings of the 2nd international conference on Embedded networked sensor systems - SenSys '04*, 2004, p. 39.
- [174] Y. Wu, S. Li, S. Liu, H.-S. Dou, and Z. Qian, *Vibration of Hydraulic Machinery*, vol. 11. Book chapter, Dordrecht: Springer Netherlands, 2013.
- [175] M. Hafida, “ANALYSE NON STATIONNAIRE DES SIGNAUX VIBRATOIRES DANS LA SURVEILLANCE DES MACHINES ET LA PREVENTION DES DEFAILLANCES,” 2013.
- [176] S. Ebersbach and Z. Peng, “Expert system development for vibration analysis in machine condition monitoring,” *Expert Syst. Appl.*, vol. 34, no. 1, pp. 291–299, Jan. 2008.
- [177] J. K. Sinha and K. Elbhah, “A future possibility of vibration based condition monitoring of rotating machines,” *Mech. Syst. Signal Process.*, vol. 34, no. 1–2, pp. 231–240, Jan. 2013.
- [178] E. P. Carden and P. Fanning, “Vibration Based Condition Monitoring: A Review,” *Struct. Heal. Monit. An Int. J.*, vol. 3, no. 4, pp. 355–377, Dec. 2004.
- [179] J. Morel, “Surveillance vibratoire et maintenance prédictive,” *Tech. l'ingénieur Comport. en Serv. des systèmes Compos. mécaniques*, vol. base docum, no. ref. article : bm5148, pp. 0–21, 2005.
- [180] S. DIOUF, “Contribution Au Diagnostic Industriel De Defauts De Roulement Et De Balourd Par Techniques Neuronales,” Université Paris-Est Créteil Val de Marne (UPEC), 2007.