



**HAL**  
open science

# Management of Dynamic Reconfiguration in a Wireless Digital Communication Context

Mohamad Al Fadl Rihani

► **To cite this version:**

Mohamad Al Fadl Rihani. Management of Dynamic Reconfiguration in a Wireless Digital Communication Context. Electronics. INSA Rennes, 2018. English. NNT: . tel-02079257v1

**HAL Id: tel-02079257**

**<https://hal.science/tel-02079257v1>**

Submitted on 25 Mar 2019 (v1), last revised 14 May 2019 (v2)

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

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

# THESE DE DOCTORAT DE

L'INSTITUT NATIONAL DES SCIENCES  
APPLIQUEES RENNES  
COMUE UNIVERSITE BRETAGNE LOIRE

ECOLE DOCTORALE N°601  
*Mathématiques et Sciences et Technologies  
de l'Information et de la Communication*  
Spécialité : *Télécommunications*

Par

« **Mohamad-Al-Fadl Rihani** »

« **Management of Dynamic Reconfiguration in a Wireless Digital  
Communication Context** »

Thèse présentée et soutenue à « INSA Rennes », le « 18 Decembre 2018 »  
Unité de recherche : IETR  
Thèse N° : 18ISAR 38 / D18 - 38

## Rapporteurs avant soutenance :

**EI-Bay BOURENNANE** Professeur, Université de Bourgogne, Dijon  
**Olivier ROMAIN** Professeur, ENSEA, Université de Cergy-Pontoise

## Composition du Jury :

**Président:** Bertrand GRANADO Professeur, UPMC, Paris  
**Rapporteur1:** EI-Bay BOURENNANE Professeur, Université de Bourgogne, Dijon  
**Rapporteur2:** Olivier ROMAIN Professeur, ENSEA, Université de Cergy-Pontoise

**Ali CHEHAB**  
Professeur, Université Américaine, AUB, Liban  
**Mohamad MROUE**  
Professeur Associé, Université Libanaise, Faculté de Génie, Liban  
**Jean-Christophe PREVOTET**  
Maître de Conférence, INSA, Rennes  
**Yasser MOHANNA**  
Professeur, Université Libanaise, Faculté des Sciences, Liban  
*Co-directeur de thèse*  
**Fabienne NOUVEL**  
Maître de Conférence, HDR, INSA, Rennes  
*Directrice de thèse*

Intitulé de la thèse :

**Management of Dynamic Reconfiguration in a Wireless  
Digital Communication Context**

**Mohamad-AI-Fadi RIHANI**

En co-direction avec :



*Document protégé par les droits d'auteur*





# Acknowledgments

First, I would like to thank God, whom blessed and gave me the strength to be what I am today.

I wish to express my sincere gratitude to the persons who helped to make my PhD year's a valuable experience and a pleasing journey that I will never forget.

I am truly grateful to my thesis supervisors, Dr. Fabienne Nouvel, and Dr. Jean-Christophe Prevotet at INSA Rennes, and for Prof. Yasser Mohanna, and Dr. Mohamad Mroue at the Lebanese University at Beirut for all their help and guidance throughout my PhD.

I would like also to thank the Lebanese University and Metro-pole Rennes for providing financial aid during the PhD.

It is also my pleasure to express my sincere thanks to the committee members, Prof. Bertrand GRANADO, Prof. EL-BAY BOURENNANE, Prof. Olivier ROMAIN and Prof. Ali CHEHAB, for reading and evaluating my manuscript.

Besides, I would like to thank the staff members in IETR, for providing a friendly working environment and facilities to complete this thesis. I also wish to express my gratitude to the officials and other staff members who rendered their help during the period of my thesis work.

My thanks also go to all my colleagues and friends at IETR, who encouraged and helped me during my stay at Rennes.

My sincere feelings of gratitude to my family, my parents, my brothers, and my sisters for their support and valuable prayers.

Finally, I thank my beloved wife, Zahraa, for her love, patience, and encouragement throughout my life.



# Abstract

Today, wireless devices generally feature multiple radio access technologies (LTE, WiFi, WiMax, ...) to handle a rich variety of standards or technologies. These devices should be intelligent and autonomous enough in order to either reach a given level of performance or automatically select the best available wireless technology according to standards availability. On the hardware side, System on Chip (SoC) devices integrate processors and FPGA logic fabrics on the same chip with fast inter-connection. This allows designing Software/Hardware systems and implementing new techniques and methodologies that greatly improve the performance of communication systems. In these devices, Dynamic Partial Reconfiguration (DPR) constitutes a well-known technique for reconfiguring only a specific area within the FPGA while other parts continue to operate independently. To evaluate when it is advantageous to perform DPR, adaptive techniques have been proposed. They consist in reconfiguring parts of the system automatically according to specific parameters. In this thesis, an intelligent wireless communication system aiming at implementing an adaptive OFDM based transmitter and performing a vertical handover in heterogeneous networks is presented. An unified physical layer for WiFi-WiMax networks is also proposed. An intelligent Vertical Handover Algorithm (VHA) based on Neural Networks (NN) was proposed to select best available wireless standard in heterogeneous network. The system was implemented and tested on a ZedBoard which features a Xilinx Zynq-7000-SoC. The performance of the system is described and simulation results are presented in order to validate the proposed architecture. Real time power measurements have been applied to compute the overhead power for the PR operation. In addition demonstrations have been performed to test and validate the implemented system.





# Résumé de la Thèse

Un système de communication sans fil est défini comme la transmission et la réception de signaux par des systèmes électromagnétiques. Il a été initialement utilisé vers 1890 en télégraphie sans fil. De nos jours, les systèmes de communication sans fil sont partout.

Le canal sans fil constitue un canal de communication des plus difficiles. Comme le canal sans fil est considéré comme variant dans le temps, il semble intéressant de disposer d'un système sans fil reconfigurable au noeud d'extrémité, comprenant certains modules de la chaîne de communication pouvant être reconfigurés pour optimiser l'utilisation de la capacité de canal disponible. Aujourd'hui encore, les utilisateurs font face à une très grande diversité de réseaux et de protocoles de communication généralement très complexes et hétérogènes. Dans ce contexte, il semble évident que nous pourrions bénéficier de l'utilisation de systèmes sans fil reconfigurables. Les questions qui sont ouvertes : comment et quand les radios reconfigurables devraient-elles être reconfigurées ? Comment décider quelle standard est le plus adapté et décider dde l'utiliser pour la communication ?

Afin de décider quand et quoi reconfigurer dans la chaîne de communication, des techniques adaptatives ont été proposées pour permettre d'appliquer des modifications en temps réel aux couches PHY et/ou MAC d'un système sans fil. Des algorithmes "d'hand-over" vertical (VHA) ont été proposés pour permettre de sélectionner le meilleur système de communication sans fil disponible sur le noeud final. Par ailleurs, les algorithmes d'apprentissage automatique sont une des techniques prometteuses utilisées dans ces VHA pour fournir plus d'intelligence au dispositif de noeud final. Lorsqu'une décision est prise, le système devra exécuter et reconfigurer la chaîne de communication. Parmi les solutions matérielles, les réseaux FPGA (Field Programmable Gate Arrays) constituent des circuits efficaces pour mettre en oeuvre de tels systèmes reconfigurables. Ces dispositifs sont maintenant suffisamment mûrs pour implémenter des systèmes très complexes avec un haut niveau de performance. La reconfiguration partielle (PR) est l'une des fonc-

tionnalités intéressantes qui ont été ajoutées par les fournisseurs de FPGA pour assurer encore plus de flexibilité. La PR fournit la possibilité de reconfigurer un module spécifique sur le FPGA pendant l'exécution, sans perturber le fonctionnement des autres blocs statiques.

Une autre caractéristique intéressante dans les récents dispositifs basés sur les FPGA est la présence de coeurs de processeurs embarqués qui sont implémentés avec la structure FPGA matérielle, dans la même puce (SoC). Les périphériques SoC sont appropriés pour la conception de systèmes logiciels et matériels (SW/HW). Il est important de comprendre et d'étudier les mécanismes d'application de la PR sur la plateforme ARM-FPGA dans le contexte d'un système sans fil reconfigurable. Quelles sont les exigences et les besoins pour créer un système sans fil reconfigurable sur les parties logicielles et matérielles ? Comment ces parties interagissent-elles entre elles et comment la reconfiguration est-elle gérée ? En outre, il est intéressant d'effectuer des mesures en temps réel pour la consommation d'énergie dans la partie PR. La gestion de la reconfiguration est étudié sur un système de communication utilisant la modulation OFDM (multiplexage par répartition de fréquence orthogonal). En tant qu'étude de cas, le passage d'une norme WIFI à WIMAX en utilisant la reconfiguration dynamique est analysé et mis en oeuvre.

Cette thèse a pour objectifs de répondre aux questions de l'intérêt, de la mise en oeuvre de la technique PR dans le contexte de systèmes de communication sans fil sur une plateforme ARM-FPGA.

## 1. Concepts et état de l'art

### 1.1 Communications sans fil

Au cours des dernières décennies, les communications sans fil sont devenues un facteur important dans notre vie quotidienne. En raison de l'essor d'Internet, les utilisateurs finaux souhaitent toujours être connectés à l'intérieur et à l'extérieur, en mobilité ou non.

Les canaux sans fil sont considérés comme des canaux les plus difficiles pour la transmission de données. Les signaux sans fil sont généralement affectés par de multiples facteurs liés à l'existence d'obstacles, de bâtiments et de la mobilité de l'utilisateur. Les chercheurs ont proposé plusieurs techniques de modulation et de multiplexage pour maximiser la capacité du canal. Parmi les différentes, celle du multiplexage par répartition de fréquences orthogonale (OFDM). Lancé

dans les années 1950, l'OFDM est une technique de multiplexage très répandue permettant une transmission de données à haut débit sur des supports filaire ou sans fil [Cha66]. L'OFDM offre de bonnes performances contre les interférences multi-trajets sur de grandes distances.

Les systèmes de communication sans fil ont beaucoup évolué en termes de mobilité, de couverture et de débit. Un grand nombre de normes avec des spécifications différentes, à la fois dans les couches PHY (bande passante, débit et codage de modulation) et MAC, sont proposées pour répondre aux différentes exigences techniques et donner accès à de nouveaux services. Les architectures les plus connues des réseaux sans fil sont les réseaux locaux sans fil (WLAN), les réseaux sans fil métropolitains (WMAN), les réseaux cellulaires et les réseaux Internet des objets (IoT). Nous pouvons par ailleurs remarquer l'existence de plusieurs normes sans fil. Cela contribuera à former des réseaux hétérogènes. Les noeuds source et destination devront savoir comment sélectionner le meilleur standard sans fil disponible en fonction des préférences requises.

## 1.2 Techniques adaptatives et radio reconfigurable

L'augmentation des performances des systèmes de communication sans fil a toujours été l'un des premiers objectifs des concepteurs. En raison des caractéristiques des canaux sans fil (perte de chemin, évanouissement, ombrage ...), de la multiplicité des normes, des attributions de fréquences et des caractéristiques de mobilité offertes par les dispositifs sans fil, l'environnement d'exploitation devient de plus en plus complexe à appréhender.

Dans ce contexte, les chercheurs ont proposé des mécanismes adaptatifs pour permettre aux systèmes sans fil d'adapter les formes d'onde en fonction des propriétés du canal. De tels systèmes analysent, apprennent et décident de se reconfigurer dynamiquement. Divers paramètres de bas niveau tels que la puissance du signal reçu (RSSI), le rapport signal / bruit (SNR), le taux d'erreur binaire (BER) ont été utilisés pour garantir l'adaptabilité. D'autres paramètres de haut niveau, tels que l'utilisation des ressources, la QoS et la consommation d'énergie, ont également été étudiés pour décider et adapter de manière appropriée le système à l'environnement en temps réel. L'un des principaux avantages des techniques adaptatives consiste à maximiser la capacité du canal tout en minimisant les coûts (exemple la consommation d'énergie).

La modulation adaptative est l'une des techniques de transmission qui a été proposée pour établir le meilleur schéma de constellation en fonction des conditions

de canal [FS], [KH00] et [MN15]. L'encodage de canal adaptatif et le taux de codage sont d'autres techniques adaptatives proposées pour sélectionner le meilleur encodeur ou le meilleur débit en fonction de l'état du canal tel que présenté dans [SPA10], [LTG04] et [DF03]. D'autres algorithmes et méthodologies ont également été proposés pour traiter l'adaptation selon la bande passante. Par exemple, dans les systèmes OFDM, les blocs de mappeur de FFT et de sous-porteuse peuvent être adaptés en fonction des paramètres fournis par le dispositif et par le réseau tels que présentés dans [ZKS07] et [FBF16]. Dans [ZEA05] et [LWN09], les auteurs montrent qu'il est important d'adapter la taille de la FFT par rapport à la consommation d'énergie. Les résultats présentés dans ces deux documents décrivent comment la consommation d'énergie augmente à mesure que la taille de la FFT augmente.

L'application de techniques adaptatives sur les communications sans fil nécessite la mise en oeuvre d'un algorithme qui détecte l'environnement et reconfigure en conséquence des blocs spécifiques dans la couche PHY pendant l'exécution, par exemple le codeur, le modulateur et les blocs IFFT.

### 1.3 Transfert vertical dans les réseaux hétérogènes

Les algorithmes "d'hand-over" vertical (VHA) ont été développés par les chercheurs pour décider quel standard sans fil dans les réseaux hétérogènes est le meilleur pour la communication. Les VHA étudient plusieurs paramètres liés au canal sans fil, au réseau, à l'application utilisateur et aux périphériques pour prendre une telle décision. Selon les valeurs des paramètres, le VHA sélectionne la meilleure norme sans fil disponible et décide de passer d'un standard sans fil à un autre si nécessaire.

Dans le contexte du "hand-over" vertical dans des réseaux hétérogènes, plusieurs travaux peuvent être mentionnés. Dans [KJU<sup>+</sup>15], les auteurs ont proposé un algorithme composé de deux étapes. Dans la première étape, le transfert est déclenché en fonction du débit demandé par l'utilisateur. Avant de débuter le transfert, le système évalue la vitesse du noeud. Si la vitesse est supérieure à un certain seuil prédéfini qui ne permet pas de bénéficier efficacement du passage à un autre système, le transfert est annulé et considéré comme inutile. Sinon, si la vitesse est appropriée, les réseaux disponibles sont comparés en fonction de cinq paramètres : la bande passante, la gigue, le retard, le coût et l'erreur binaire. Ensuite, le meilleur standard est sélectionné et le changement de standard est effectué si nécessaire.

Les VHA proposés les plus prometteurs sont ceux basés sur les techniques d'apprentissage automatique. Ils peuvent être catégorisés comme des systèmes basés sur l'intelligence définie dans [ABG14] avec les autres VHA basés sur la logique floue. Les VHA détectent généralement les standards sans fil disponibles et prennent une décision basée sur les valeurs actuelles de certains paramètres. D'un autre côté, les VHA basés sur des techniques d'apprentissage automatique auront la capacité d'étudier et d'apprendre des étapes précédentes pour prendre une décision efficace.

Les réseaux de neurones (NN) sont des systèmes informatiques inspirés des réseaux de neurones biologiques qui constituent le cerveau. De tels systèmes d'apprentissage sont considérés comme un algorithme d'apprentissage automatique supervisé. Le réseau de neurones apprend à partir d'un ensemble de données pour générer des sorties. L'apprentissage par renforcement est un domaine d'apprentissage automatique qui concerne la façon dont les agents doivent agir dans un environnement afin de maximiser le résultat. Il est similaire à la programmation dynamique, dans laquelle les résultats sont enregistrés après chaque action et la fonction est reconfigurée pour maximiser le résultat. Les VHA basés sur des algorithmes d'apprentissage automatique ont fourni de bonnes performances et ont la capacité d'apprendre des anciens enregistrements pour créer des fonctions de prédiction.

L'une des principales étapes du "hand-over" vertical est l'étape d'exécution. Après avoir décidé de passer d'un standard sans fil à un autre, au niveau du noeud d'extrémité, la communication doit être commutée de la couche MAC / PHY actuelle à la couche MAC / PHY cible. Certaines études considèrent dès lors la création d'une chaîne reconfigurable unifiée pour plusieurs normes sans fil.

## **1.4 Plates-formes reconfigurables flexibles pour les systèmes de communication**

Contrairement à un ASIC, un FPGA est un circuit intégré conçu pour être configuré par un concepteur. Les dispositifs FPGA contiennent un ensemble de blocs logiques programmables et une hiérarchie d'interconnexions reconfigurables qui permettent aux blocs d'être interconnectés entre eux. La reconfiguration partielle est une fonctionnalité intéressante qui a été rendue disponible dans certaines familles de FPGA. Il offre la possibilité de reconfigurer un sous-ensemble de blocs logiques dans le FPGA sans altérer le fonctionnement des autres blocs implantés.

Au cours de la dernière décennie, la recherche sur la conception de micropro-

cesseurs intégrés et d'architectures hybrides FPGA est devenue un sujet d'intrêt. L'idée consiste à combiner les dispositifs FPGA traditionnels et les microprocesseurs avec des périphériques associés afin de construire un système complet sur puce programmable (SoPC). A titre d'exemple, le Zynq 7000 proposé par Xilinx est un composant SoC dont l'architecture est constituée d'un processeur ARM Cortex A9 dual-core et d'un FPGA Artix-7 ou Kintex-7 [CEES14].

Les systèmes d'exploitation pour les systèmes numériques embarqués sont généralement conçus pour être efficaces en ressources et en énergie. Le concept de base de micro-noyau a été proposé dans [Han70] comme un noyau réduit avec des mécanismes d'ordonnancement fondamentaux. Les micro-noyaux s'exécutent sur la partie processeur (PS) et peuvent utiliser des blocs sur FPGA en tant qu'accélérateurs matériels. L'un de ces micro-noyaux proposés est Ker-ONE [XPN15]. Ker-ONE est très simple, il ne fournit que des fonctions fondamentales telles que l'ordonnancement circulaire, la communication interprocessus et la gestion de la mémoire.

## **1.5 Application de la reconfiguration partielle dans le contexte de radios reconfigurables**

En raison de ses propriétés intéressantes, il semble très intéressant de bénéficier de la technique de reconfiguration partielle dans le contexte des radios reconfigurables, car elle permet de reconfigurer les modules matériels lors de l'exécution. De nombreux travaux proposés pour les systèmes de communication sans fil reconfigurables et la radio cognitive ont déjà mis à profit la technique de reconfiguration partielle dans les FPGA.

## **2. Plate-forme ARM-FPGA pour systèmes de communication sans fil adaptatifs automatisés utilisant une technique de reconfiguration partielle**

Dans cette section, nous présentons une plate-forme logicielle / matérielle pour les systèmes de communication sans fil adaptatifs automatisés. Le système proposé bénéficie de la technique de reconfiguration partielle disponible dans le dispositif FPGA. Les fonctions de bande de base du système sans fil sont implémentées dans la structure FPGA, tandis que la reconfiguration des différents blocs fonctionnels

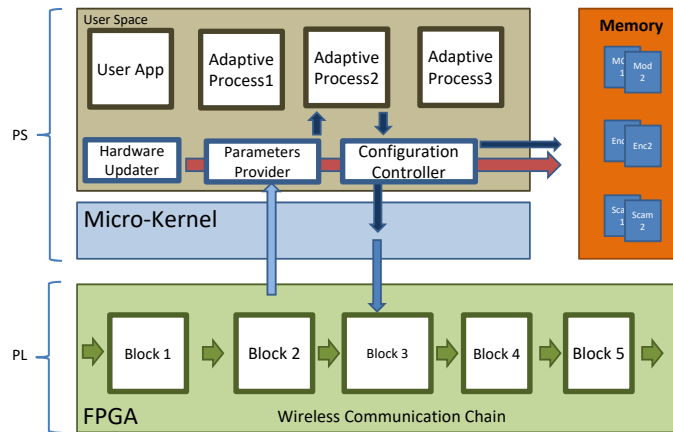


FIGURE 1 – Blocs de conception générale du système

est gérée par des processus s'exécutant sur le processeur ARM. Les processus sont basés sur des algorithmes adaptatifs. Leur rôle est de prendre des décisions et de gérer le processus de reconfiguration des modules concernés.

## 2.1 Conception générale du système

Le système SoC proposé peut être divisé en deux parties : la première partie est exécutée sur le processeur ARM, tandis que la seconde partie est exécutée dans le FPGA. La conception générale du système est présentée dans la FIGURE 1.

Les modules proposés sur l'ARM sont les suivants : le Contrôleur de configuration, le fournisseur de paramètres et le module de mise à jour matérielle. Tous ces modules fonctionnent comme des tâches dans l'espace utilisateur du micro-noyau Ker-One. Le contrôleur de configuration doit appliquer et contrôler la PR sur le PL et contrôler le flux de données pendant l'opération de reconfiguration partielle. Le fournisseur de paramètres fournit aux processus adaptatifs les paramètres nécessaires à la prise de décision. Le processus de mise à jour matérielle ajoute ou met à jour les nouvelles versions des modules reconfigurables.

Sur la partie PL, une chaîne de communication reconfigurable (niveau PHY) est implémentée en utilisant la conception reconfigurable dynamique. Ainsi, la chaîne de communication sans fil est divisée en parties statiques et reconfigurables



dynamiquement. Les modules reconfigurables sont reconfigurés pendant l'exécution lorsque le processus adaptatif le décide.

## 2.2 Émetteur OFDM adaptatif automatique

Pour étudier l'impact de l'application de la reconfiguration partielle dans le contexte des radios reconfigurables, nous avons implémenté un transmetteur OFDM auto-adaptatif.

L'émetteur OFDM générique est composé d'un codeur, d'un modulateur, d'un mappeur et d'une IFFT. Sur la base de l'outil Xilinx Vivado, plusieurs bitstreams partiels ont été générés pour ces modules.

Dans la conception PR, les différentes versions des blocs reconfigurables seront stockées en mémoire et chargées à la demande. Si le mécanisme de reconfiguration partielle n'est pas mis en oeuvre, il convient alors d'implanter toutes les versions possibles de chacun des blocs. Ceux-ci seront alors rendus actifs selon la demande (choix des paramètres du standard) et le flux de données y est alors appliqué. Ce type d'implantation est qualifié d'implantation parallèle. En utilisant la PR, il est certain que les ressources utilisées sur le FPGA diminuent, au fur et à mesure que le nombre de blocs implémentés diminuent.

L'outil Xilinx SDK a été utilisé pour implémenter trois processus adaptatifs s'exécutant sur le processeur ARM. Les processus sont implémentés en code C et s'exécutent parallèlement au micro-noyau Ker-One. Dans notre scénario, nous appliquerons trois processus adaptatifs. Le premier processus concerne la modulation adaptative qui décide de reconfigurer le module modulateur. Le second processus est le codage adaptatif qui contrôle le module codeur et le reconfigure selon les conditions de canal. Le troisième processus adaptatif est le processus IFFT adaptatif qui reconfigure le module IFFT.

## 2.3 Résultats expérimentaux

Cette section présente les résultats obtenus à partir du scénario d'émetteur OFDM adaptatif. Les résultats analysent le temps nécessaire pour reconfigurer les modules dynamiques, la taille des bitstreams partiels, la consommation d'énergie et les ressources réservées par chaque version de modules. Enfin, un diagramme de Gantt illustre le comportement du système en fonction des variations des paramètres liés aux demandes de reconfiguration.

Comme décrit précédemment, nous considérons trois modules de la chaîne OFDM

reconfigurables : le modulateur, l'encodeur et l'IFFT. Les résultats présentés dans TABLE 1 incluent la taille des bitstreams partiels et le temps requis pour reconfigurer chaque module respectivement.

TABLE 1 – Taille partielle de Bitstream et temps de reconfiguration

Module	Taille partielle de Bitstream	Temps de reconfiguration
IFFT	188 KB	1.48 ms
Encoder	106 KB	0.834 ms
Modulation	30 KB	0.236 ms

Comme indiqué dans la TABLE 1, le temps nécessaire pour transférer un flux de données vers le matériel à l'aide de l'interface PCAP du SoC dépend de la taille du bistream.

En outre, il est important d'analyser les processus en cours à travers un diagramme de Gantt tel que représenté dans la FIGURE 2. Comme le montre cette figure, trois processus adaptatifs fonctionnent en parallèle sur la partie PS, ils analysent leurs propres paramètres et détectent finalement tout changement. Si nécessaire, la reconfiguration partielle est appliquée sur le bloc approprié afin d'adapter le système au nouvel environnement.

On peut remarquer que le temps nécessaire pour reconfigurer le module IFFT est plus long que celui des modules de codage et de modulation. Cela est dû à la différence dans la taille des bistreams partiels. Pendant la reconfiguration partielle, les modules précédents dans la chaîne sont "suspendus" comme indiqué dans la FIGURE 2. Le temps de suspension dépend de la taille des bistreams partiels transférés. Comme illustré, la suspension d'exécution n'affecte que les modules précédents le bloc reconfiguré.

Avec la reconfiguration partielle, la consommation d'énergie du FPGA et les ressources réservées sont réduites. En alternative à cette technique, toutes les versions des différents modules peuvent être implémentées dans le FPGA, avec des multiplexeurs supplémentaires pour sélectionner le module souhaité. Cela nécessite généralement beaucoup plus de ressources sur le FPGA. En conséquence, la consommation d'énergie statique et dynamique de la puce serait relativement plus élevée vis à vis d'un système basé sur la PR.

La TABLE 2 indique la quantité de ressources utilisées lors de l'application de la technique PR par rapport au cas où la PR n'est pas adoptée. On peut voir qu'il y a environ un rapport de 4 entre les deux cas. De plus, dans le cas de la PR, le gain de ressources permet d'utiliser les parties restantes du FPGA pour implémenter

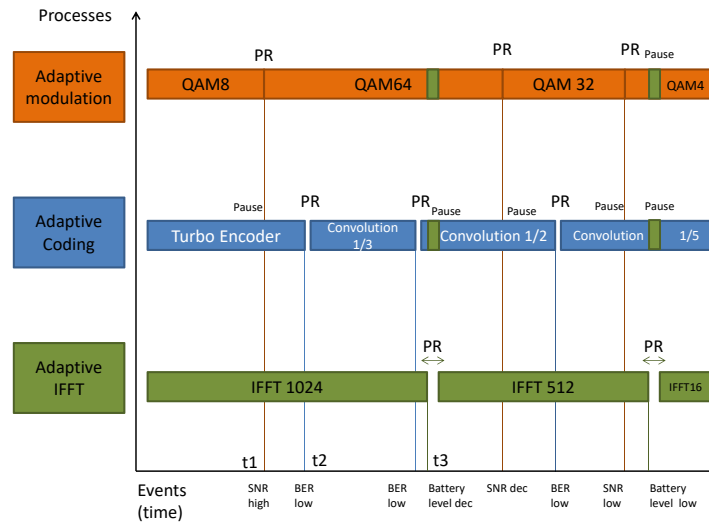


FIGURE 2 – Processus adaptatifs reconfigurant l'émetteur OFDM pendant l'exécution en fonction des valeurs des paramètres

TABLE 2 – Comparaison de l'utilisation des ressources matérielles

Ressources	LUTs	REGs	DSPs
avec reconfiguration partielle	2578	4129	12
sans reconfiguration partielle	10320	16938	57
Disponible sur ZYNQ	53200	106460	220

des unités de traitement supplémentaires. Le pourcentage des LUT, REG et DSP utilisés sur le SoC est respectivement de 19,3%, 15,9% et 25,9%, lorsque tous les blocs sont implémentés en parallèle. Lors de l'utilisation de la technique PR, le pourcentage de ressources utilisées est respectivement de 4,8%, 3,8% et 5,4%.

### 3. Couche PHY unifiée reconfigurable pour les systèmes sans fil utilisant la reconfiguration partielle - VHA

Dans cette section, le principe de la reconfiguration partielle appliquée à différents standards est étudié. Deux architectures d'implémentation différentes et

reconfigurables sur PL sont comparées. Une chaîne sans fil reconfigurable unifiée pour Wifi-WiMax est testée en tant qu'étude de cas pour évaluer les résultats expérimentaux.

### **3.1 Plate-forme ARM-FPGA pour la commutation entre plusieurs systèmes sans fil**

Le système étudié est divisé en deux parties : logiciel et matériel. Dans la partie logicielle, les processus s'exécutent sur le processeur ARM pour contrôler et gérer la configuration de la chaîne sans fil implémentée sur le matériel. Dans la partie matérielle, les systèmes sans fil reconfigurables sont implémentés et conçus pour être partiellement reconfigurés pendant leur exécution.

La partie logicielle du système s'exécute dans l'espace utilisateur du micro-noyau léger dédié à la reconfiguration partielle. Le rôle du VHA est de lire les paramètres des systèmes de communication disponibles, d'apprendre et de décider s'il faut passer à un autre système ou continuer à fonctionner avec le système actuel.

Le flux de données de la partie logicielle peut être résumé comme suit : Le fournisseur de paramètres fournit au VHA les paramètres pertinents des différentes couches d'u standard. Le VHA analyse les valeurs des paramètres et prend une décision concernant un hand-over possible. En conséquence, une demande de hand-over est envoyée par le VHA au contrôleur de configuration. Le contrôleur de configuration gère le transfert des bitstreams vers les modules reconfigurables du FPGA à travers l'interface PCAP.

La partie matérielle du système implante les couches physiques des systèmes de communication mis en place sur le FPGA. En utilisant la reconfiguration partielle, nous visons à mettre en oeuvre une seule chaîne reconfigurable unifiée pour différentes normes sans fil. Le système de communication est supposé fonctionner en parallèle avec d'autres accélérateurs matériels sur le FPGA. Lorsque la reconfiguration partielle est appliquée, les autres accélérateurs ne sont pas impactés.

### **3.2 Architectures reconfigurables proposées**

Dans la partie PL, une seule couche physique reconfigurable pour plusieurs systèmes sans fil est implémentée.

Dans la première architecture, toute la chaîne de communication est considérée comme un module reconfigurable. Dans ce cas, chaque standard sans fil ne possède

qu'un seul bistream partiel. Lorsque le système décide de passer à un autre système sans fil, le bistream partiel souhaité est transféré et remplace l'existant. Cette architecture reconfigurable est dénommée une architecture de bloc reconfigurable (ORBA).

La deuxième architecture divise le système de communication en plusieurs blocs reconfigurables. Dans ce cas, pour chaque bloc reconfigurable dans la chaîne,  $n$  bistreams partiels sont stockés en mémoire, où  $n$  est le nombre de normes sans fil considérées. Si chaque chaîne est divisée en  $m$  blocs reconfigurables, alors  $(m * n)$  bistreams sont nécessaires. Cette architecture reconfigurable est appelée architecture à blocs reconfigurables multiples (MRBA).

Les deux schémas sont comparés pour définir les avantages et les inconvénients de chaque architecture.

### 3.3 Étude de cas : Chaîne reconfigurable unifiée WiFi-WiMax

Tout d'abord, la couche physique pour chaque norme est présentée, puis nous expliquons la chaîne unifiée.

IEEE définit la couche PHY et les spécifications des couches MAC pour WIFI et WIMAX. La couche physique de ces deux normes est basée sur le schéma de modulation OFDM. Les modules principaux sont : Brouilleur/Débrouilleur, Codeur/Décodeur de Correction d'Erreur Directe (FEC), Interleaver/De-Interleaver, Mapper/De-Mapper, Insertion/Retrait de Pilot Guard, IFFT/FFT et Insertion/-Suppression de Préfixe Cyclique. Les différences dans les architectures des émetteurs sont illustrées dans la FIGURE 3. Comme indiqué dans la FIGURE 3, les architectures de couche physique des deux standards sont assez similaires avec peu de différences dans la fonctionnalité de certains modules. Par exemple, WIMAX utilise un FEC Reed-Solomon avec un encodeur de convolution tandis que WIFI utilise uniquement l'encodeur de convolution. Les différences dans le récepteur sont similaires à celles de l'émetteur. Sur cette base, il semble approprié d'appliquer la PR avec la méthode MRBA.

L'application de la PR dans la conception du système à l'aide de MRBA nécessite de diviser le système en deux parties, statique et dynamique. La partie statique est constituée de blocs partagés utilisés dans les deux normes. Pendant le processus de commutation entre les normes, cette partie du système reste inchangée. La partie dynamique est composée de modules qui peuvent être reconfigurés si nécessaire. Plusieurs versions de ces modules reconfigurables sont stockées en mémoire.

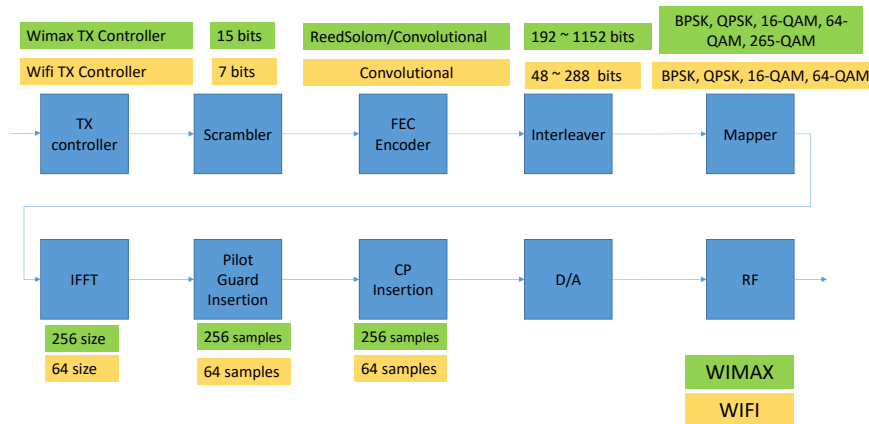


FIGURE 3 – Liste des modules reconfigurables pendant VHA

### 3.4 Résultats expérimentaux

Les résultats expérimentaux pour la chaîne unifiée de Wifi et WiMax sont donnés en terme de temps de reconfiguration, de ressources sur FPGA, et de consommation sur puce.

Les outils d'estimation de puissance et Vivado ont été utilisés pour estimer la consommation d'énergie et les ressources réservées des différentes versions des modules reconfigurables. Comme indiqué dans la TABLE 3, l'application de la reconfiguration partielle permet de réduire la consommation d'énergie et les ressources utilisées sur le FPGA. Pour l'émetteur, les résultats montrent que les ressources

TABLE 3 – Comparaison de l'utilisation des ressources matérielles

	LUTs	REGs	DSPs
chaîne unifiée pour les deux normes	2229	3376	9
les deux normes mises en oeuvre	3967	6027	15
Disponible sur ZYNQ	53200	106460	220

utilisées sont réduites d'environ un facteur de 1,7, lorsque la chaîne unifiée partiellement reconfigurable est utilisée. Le pourcentage de ressources utilisées lorsque les deux normes sont implémentées est respectivement de 6,9%, 5,6% et 6,81% pour

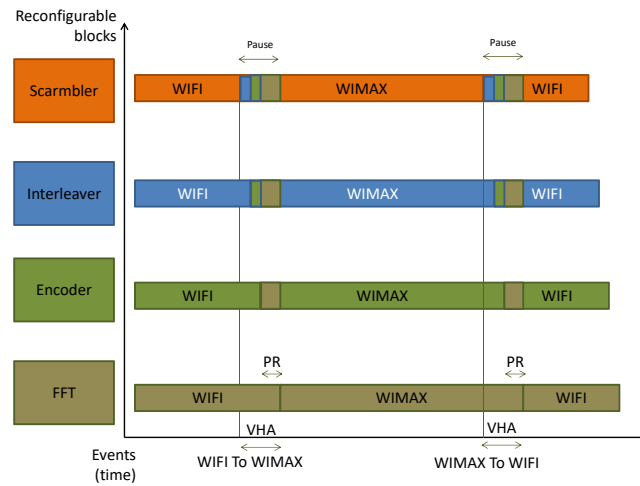


FIGURE 4 – Etat des modules reconfigurables pendant le VHA

les LUT, les REG et les DSP. En revanche, le pourcentage est de 4,18%, 3,17% et 4,09%. lorsqu'une chaîne unifiée est utilisée pour les deux normes.

Dans la FIGURE 4, un diagramme de Gantt illustre un scénario qui montre l'état des modules reconfigurables implémentés sur FPGA lors de l'application du VHA. Le graphique est établi en fonction des temps de reconfiguration mesurés et reflète la manière dont la chaîne unifiée passe d'un standard à l'autre. Comme le montre la figure, les modules sont reconfigurés séquentiellement dans un ordre lié à leur position dans la chaîne. Le temps nécessaire pour passer d'un standard à un autre est la somme des temps de reconfiguration requis par chaque module.

Comme indiqué, lorsqu'un module est reconfiguré, le flux de données est mis en pause jusqu'à ce que les modules suivants soient également reconfigurés. Comme mentionné précédemment, le temps nécessaire pour reconfigurer un bloc est lié à sa taille. Ceci est illustré dans la FIGURE 4 dans laquelle le temps nécessaire pour reconfigurer le bloc IFFT est supérieur à celui requis par les autres. Le passage de WIFI à WIMAX est réalisé lorsque le dernier module (par exemple IFFT) est reconfiguré avec succès. Enfin, les données sont transférées via la norme WIMAX jusqu'à ce qu'une autre VHA se produise.

## 4. VHA intelligent pour les réseaux sans fil utilisant les réseaux neuronaux et l'apprentissage par renforcement

Dans les réseaux hétérogènes, les utilisateurs mobiles peuvent détecter de multiples normes sans fil telles que l'accès sans fil haut débit mobile (3G, 3.5G et 4G) et les technologies WLAN. Dans cette section, nous présenterons l'application du VHA pour les réseaux hétérogènes dans lesquels le WiFi et le WiMax sont présents. Ce VHA est considéré comme l'agent de décision dans la plateforme ARM-FPGA proposée.

### 4.1 VHA pour WIFI et WiMax basé sur le poids des paramètres

Le VHA proposé pour un réseau WIMAX-WIFI se déroule en trois étapes : un déclencheur de transfert/hand-over, une décision initiale, et une décision finale. L'algorithme identifie deux contextes d'exécution. Dans le premier contexte, le système sans fil communique via le standard WIMAX, alors que dans le second cas, le système fonctionne initialement avec WIFI.

Le déclenchement est initié dès que le processus de transfert détecte une valeur  $RSS\_WIFI$  supérieure au  $RSS\_WIMAX$ . L'état de décision initiale est utilisée pour prendre des décisions rapides et détecter de faux transferts. un faux transfert est détecté si la vitesse du mobile est supérieure à un seuil  $WIFI\_max\_speed\_limit$ . Si la décision initiale est validée, le système passe à l'étape de décision finale. Dans cette dernière étape, un système de notation adaptatif lit plusieurs paramètres pour décider quel standard est le meilleur selon les préférences de l'utilisateur. L'ajout de l'adaptabilité au système de notation améliore l'efficacité du système.

### 4.2 VHA proposé basé sur l'apprentissage automatique

Il est intéressant d'utiliser les techniques de ML (Machine learning) pour rendre le VHA plus intelligent. L'apprentissage automatique fait référence à la science consistant à faire en sorte que les ordinateurs agissent sans être explicitement programmés. Les Neural Networks (NN) sont l'une des techniques ML intéressantes. De tels systèmes d'apprentissage sont considérés comme un algorithme d'apprentissage automatique supervisé. Les systèmes NN apprennent à partir d'un ensemble



de données pour générer des sorties, des décisions, des classes.

L'apprentissage par renforcement est un domaine d'apprentissage automatique qui traite de la manière dont les agents logiciels doivent agir dans un environnement afin de maximiser la récompense. Il est similaire à la programmation dynamique, où les récompenses sont enregistrées après chaque action et la fonction du système est reconfigurée pour maximiser la récompense.

Dans ce travail, nous proposons un système qui combine à la fois des algorithmes d'apprentissage automatique supervisés et non supervisés pour passer d'une norme à l'autre. Ce mécanisme qui est implémenté dans le système est basé sur une amélioration du processus de notation adaptative proposé précédemment. L'idée principale est de bénéficier du travail précédent dans une première étape. Cette étape consiste à collecter des enregistrements et des récompenses/retours du système de notation adaptatif précédent. Les décisions prises sont ensuite classées en fonction des récompenses obtenues. Dans un deuxième temps, seules celles qui génèrent de bonnes récompenses sont considérées et elles formeront un nouvel ensemble de données. Les enregistrements collectés et les récompenses acceptables ou bonnes correspondantes seront utilisés pour l'apprentissage du réseau de neurones NN.

Après la collecte d'un grand ensemble de données, le réseau de neurones NN est considéré comme suffisamment "formé" et stable pour prendre une bonne décision. Ainsi, il peut être adopté pour prendre les décisions. Le système peut répéter la procédure d'apprentissage du NN après la mise à jour de l'ensemble de données afin d'optimiser cette opération et d'améliorer son comportement. Le système proposé est constitué des blocs représentés sur la FIGURE 5.

### 4.3 Résultats expérimentaux

Pour prouver le concept, une simulation a été réalisée sur la base d'un scénario WiFi-WiMax utilisant un simulateur Java. Basé sur le système de notation adaptative et un système de récompense/retour simple, un simulateur Java a été mis en place pour générer les données requises (Paramètres d'entrée - Décisions - Récompenses/Retours - et Décisions d'apprentissage). Le simulateur génère aléatoirement les paramètres d'entrée : RSS, Charge réseau, Préférences utilisateur, Niveau de puissance et Statut. Ensuite, les décisions et récompenses associées pour la puissance et la QoS peuvent être sélectionnées selon une décision simple et un moteur de récompenses. Le moteur fonctionne en fonction des paramètres d'entrée et des décisions précédentes.

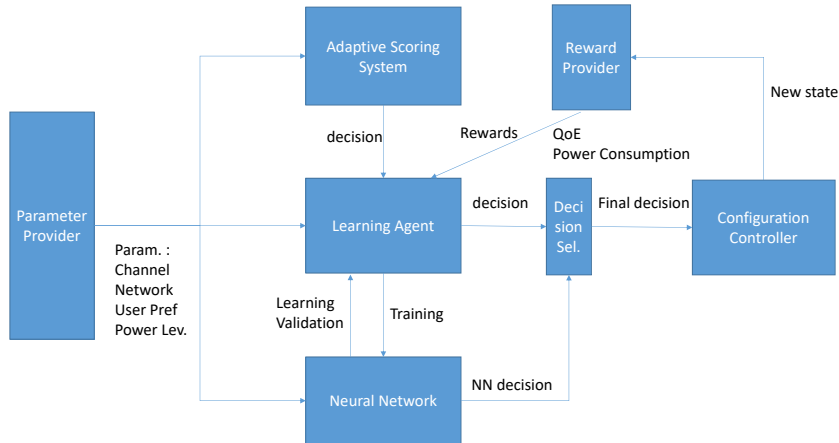


FIGURE 5 – Conception proposée de VHA

Comme nous l'avons déjà expliqué, l'agent d'apprentissage doit savoir quand il faut que le NN apprenne. Il a également pour tâche de valider le NN chaque fois qu'il y a suffisamment d'enregistrements étiquetés. Il est important de savoir quel est le meilleur nombre d'enregistrements étiquetés nécessaires pour obtenir de bons résultats.

Le taux de décisions réussies conduit à en ajouter une pour une bonne décision et en soustraire une pour une mauvaise décision. Les résultats prennent en compte le retour de la consommation d'énergie pour décider si les décisions obtenues sont bonnes ou mauvaises. Il est à souligner que la formation du NN avec les enregistrements marqués qui représentent les enregistrements avec seulement de bonnes récompenses donne de meilleurs résultats. Le même scénario a été appliqué en prenant en considération la récompense QoS comme récompense principale pour décider de la qualité de la décision prise.

## 5. Implémentation et mesures de puissance en temps réel

Dans cette section, nous présentons la implémentation du PR sur les outils Vivado pour la technique de modulation adaptative dans une chaîne OFDM. Ensuite, nous présentons les résultats de consommation d'énergie sur le SoC. À la fin de ce chapitre, deux démonstrations sont présentées. La première concerne le déroulement des processus du micro-kernel Ker-One et de la PR. La deuxième démonstration concerne la commutation entre le WiFi et le WiMax en utilisant la technique PR.

### 5.1 Implémentation et analyse de puissance pour un transmetteur OFDM reconfigurable automatisé et adapté

La chaîne générique de l'émetteur SISO OFDM est formée par différentes entités. Ces entités sont implémentées en utilisant VHDL. Certaines de ces entités sont créées à l'aide du générateur IP Xilinx. Ils sont connectés par mappage de port pour former la chaîne complète. Le code VHDL du transmetteur SISO OFDM a été implémenté à l'aide de la suite de conception Xilinx ISE et testé avec le simulateur Modelsim. Les résultats générés par le simulateur Modelsim ont été comparés à ceux obtenus avec Matlab pour garantir et valider la fonctionnalité de l'ensemble du système.

Le flux de conception Xilinx pour la reconfiguration partielle est utilisé pour mettre en oeuvre le concept de reconfiguration sur la chaîne de communication après avoir testé les différents blocs de code VHDL. Pour mettre en oeuvre la technique de modulation adaptative, le concept PR a été appliqué sur le bloc de modulation. Dans ce cas, les blocs Encoder, Mappeur, Bloc IFFT, Scaler et FIFO constituent les parties statiques du système. En utilisant Xilinx ISE ou Vivado Tools, le bloc de modulation a été resynthétisé sous forme de boîte noire. Une boîte noire est une implémentation d'une entité avec une définition de ports mais sans architecture. Le système a été rendu auto-intelligent en appliquant le concept DPR (Dynamic PR) sur la partie PL du Soc. Pendant la phase de démarrage, le système envoie le bistream principal au FPGA, puis configure le PS. Il charge également les bistreams partiels en mémoire permettant au PS d'y accéder lors de l'exécution de la reconfiguration.

La puissance consommée par la puce FPGA est la somme des consommations

d'énergie statique et dynamique. La puissance statique est principalement due au courant de fuite des transistor. Cela se produit même lorsque le transistor est "off" logiquement. La puissance dynamique est la somme de la consommation d'énergie transitoire et de la consommation d'énergie de la charge capacitive. Dans notre cas, nous utiliserons Xpower Analyzer, un analyseur de phase de conception, pour estimer les consommations électriques statiques et dynamiques. Les résultats montrent un taux de réduction de 1,6% de la puissance totale sur la puce et un taux de réduction de la puissance dynamique de 18,7%.

## 5.2 Analyse dynamique et partielle de la consommation d'énergie lors de la reconfiguration des mesures d'exécution pour les appareils SoC Zynq

Il est intéressant dans notre travail de calculer la consommation d'énergie pour l'opération de reconfiguration partielle dynamique à la fois sur le PS et le PL.

Dans cette étude, les cartes d'évaluation ZC70 ZYNQ et ZC702 ZYNQ ont été utilisées. La carte ZC706 possède un SoC AP-SoC ZYNQ 7000-XC7Z045, alors que la carte ZC702 est équipée d'une carte SoC AP ZYNQ 7000-XC7Z020 CLG484. Le SoC ZYNQ 7000-XC7Z045 fournit plus de ressources que le ZYNQ 7000-XC7Z020. Dans les deux dispositifs ZYNQ, les parties PS et PL sont intégrées avec des interfaces d'interconnexion AMBA AXI à bande passante élevée.

Les deux contrôleurs de consommation sont accessibles via l'interface I2C en utilisant le protocole de commandes PMBus [pmb]. Nous avons utilisé l'outil TI pour obtenir une vue d'ensemble de la consommation d'énergie sur PL et le PS lors d'une reconfiguration partielle. De plus, nous exécutons notre code implémenté sur le processeur pour contrôler les données à envoyer, et pour lire ces données uniquement pendant le processus de reconfiguration partielle.

La puissance consommée par le FPGA est mesurée sur deux SoC ZYNQ de taille différente. L'objectif est d'évaluer l'impact de la taille de la partie logique sur la consommation d'énergie pendant la reconfiguration partielle.

Pour la partie PS, le processus DPR est comparé à un processus de référence dans lequel la reconfiguration partielle n'est pas appliquée. Par conséquent, le pilote DevCfg avec l'interface PCAP n'est pas utilisé. Ce processus de référence consiste à exécuter une boucle while infinie et vide. Le processus DPR est un code C exécuté sur le processeur et a accès au pilote DevCfg pour appliquer des transferts de bits partiels consécutifs au PL. Dans ce cas, l'interface PCAP est engagée dans le

processus PR et le processeur ARM envoie des requêtes répétitives pour effectuer la reconfiguration partielle sur le bloc concerné. La puissance consommée par le VCCPINT est mesurée à l'aide de l'outil TI GUI.

La puissance consommée par le processus de référence est légèrement supérieure à celle obtenue pour le processus DPR. Ce résultat montre qu'il n'y a pas d'augmentation de la consommation d'énergie pour le processus de reconfiguration partielle sur le processeur. De plus, nous avons mesuré la puissance consommée par le régulateur VCCPAUX pendant la DPR et le processus de référence. Les résultats montrent qu'il y a une légère puissance supplémentaire consommée pendant le processus DPR. Les valeurs moyennes pour les mesures de puissance présentées dans les figures précédentes sont résumées dans TABLE 4.

Power Reg	with DPR	No DPR
VCCINT	274,8 mW	274,4 mW
VCCPINT	40.2 mW	41.8 mW
VCCPAUX	170.3 mW	157.7 mW

TABLE 4 – Puissance moyenne pour PS et PL dans deux cas

### 5.3 Démonstrations

Dans cette section nous présenterons deux démonstrations, la première démonstration illustre le fonctionnement du micro-noyau Ker-ONE utilisé pour le contrôle de la reconfiguration. La deuxième démonstration décrit l'application du VHA entre le WiFi et le WiMax basée sur une reconfiguration partielle.

La première démonstration est exécutée sur la plateforme Xilinx Zynq-7000 SoC, avec le processeur ARM Cortex-A9 dual-core et le FPGA reconfigurable. Plusieurs algorithmes de communication sont implémentés dans le FPGA. Ils sont considérés comme des accélérateurs matériels vis à vis des applications s'exécutant sur l'ARM, par exemple la FFT. Au dessus de Ker-ONE, plusieurs OS( ucos-II) s'exécutent simultanément en mode machines virtuelles (OS invités) avec des priorités différentes. La démonstration montre comment Ker-One traite les requêtes des machines virtuelles selon leur priorité et les demandes de reconfiguration sur le FPGA.

Dans la seconde démonstration, nous proposons un système implémentant le VHA pour les standards WIFI-WiMAX sur la même plateforme utilisant la technique PR. Le VHA fonctionne sur l'ARM. La chaîne reconfigurable unifiée qui

implémente les couches PHY des deux normes sans fil est implémentée dans la partie PL du SoC. La démonstration est constituée en deux parties. La première partie consiste en une application en code C fonctionnant sur le processeur ARM avec la chaîne de communication sans fil implémentée sur le FPGA. La deuxième partie est un code JAVA exécuté sur le PC pour simuler les paramètres de prise de décision et gérer l'interface graphique. Le simulateur sur le PC et l'application sur ARM communique via l'UART. Le simulateur collecte les valeurs de paramètres générées à partir de modèles mathématiques et les transmet dans un paquet à l'application sur l'ARM en temps réel. L'application sur l'ARM extrait les valeurs et applique le VHA pour sélectionner le meilleur standard. Si le VHA décide de passer d'un standard à un autre, il demandera au contrôleur de reconfiguration de reconfigurer la chaîne unifiée. Sur la ZedBoard, l'application transmet le nouveau statut des modules reconfigurables au simulateur. Cette transmission est également illustrée dans l'interface graphique (voir FIGURE 6).

La démonstration illustre ainsi l'effet de la mobilité d'un utilisateur, la décision prise par le VHA et l'état des blocs reconfigurables sur le FPGA.

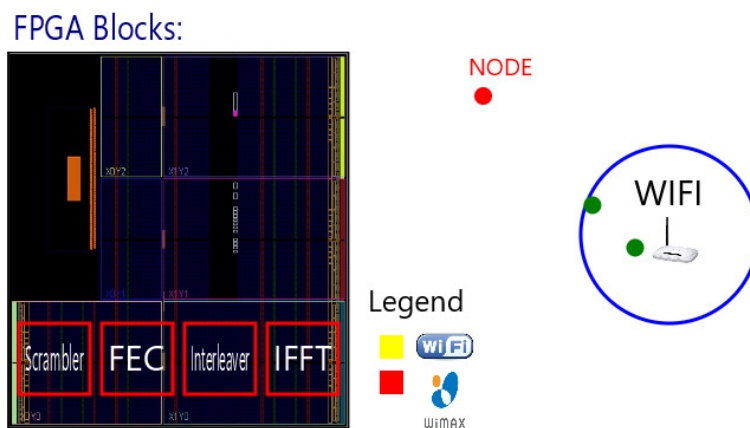


FIGURE 6 – Capture d'écran du simulateur illustrant l'état des blocs reconfigurables, le noeud mobile et la décision VHA

## 6. Conclusion et perspectives

Au cours de la dernière décennie, les systèmes de communication sans fil ont beaucoup évolué en termes de mobilité, de couverture et de débit. Un grand nombre de normes avec des spécifications différentes, à la fois dans la PHY (bande passante, débit, et techniques de codage de modulation) et MAC couches, ont été fournis pour répondre aux différentes exigences techniques et donner accès à de nouveaux services.

La reconfiguration matérielle des systèmes de communication sans fil est la solution la plus intéressante pour mettre en oeuvre des techniques adaptatives et VHA. Dans ce cas, certaines fonctions du système peuvent être modifiées pendant l'exécution, tandis que d'autres continuent à fonctionner sans interruption. Les FPGA constituent les circuits idéaux pour implémenter de tels systèmes reconfigurables. De tels dispositifs sont maintenant suffisamment mûrs pour implémenter des systèmes très complexes avec un haut niveau de performance. PR est l'une des fonctionnalités intéressantes qui ont été ajoutées par les fournisseurs de FPGA pour assurer encore plus de flexibilité.

Une autre caractéristique intéressante dans les récents dispositifs FPGA est la présence de coeurs de processeurs embarqués qui sont implémentés avec la structure FPGA matérielle, dans la même puce. Par exemple, les appareils SoC Zynq de Xilinx sont équipés d'un processeur ARM Cortex-A9 à double coeur (PS) ainsi que d'un PL dans un seul appareil. Comparés à leurs homologues de processeurs soft core, ces processeurs offrent beaucoup plus de puissance et de rapidité de calcul. Les appareils SoC sont appropriés pour la conception de systèmes logiciels et matériels communs. L'existence de tels systèmes dans les terminaux permet de concevoir des objets de communication intelligents haute performance.

Dans cette thèse, nous présentons une plate-forme ARM-FPGA pour les systèmes de communication sans fil reconfigurables qui bénéficie de la technique de reconfiguration partielle. Le système proposé a été testé pour la chaîne OFDM auto-adaptative. Il a également été utilisé pour basculer entre les réseaux WiFi et WiMax basés sur l'apprentissage automatique VHA. Des mesures de puissance et des démonstrations en temps réel ont été effectuées pour extraire les résultats expérimentaux et tester le système proposé.

Pour le travail à venir, il est intéressant d'avoir un vrai test pour une communication sans fil entre deux ou plusieurs cartes basées sur SoC avec l'application de PR pour reconfigurer les blocs dans la chaîne sans fil. Il sera également intéressant d'appliquer un test réel pour une commutation entre deux standards sans

fil sur deux cartes ARM-FPGA en utilisant une technique de reconfiguration partielle. Il sera également intéressant de proposer une architecture combinée entre PR et conception parallèle pour optimiser la meilleure puissance et le temps de reconfiguration pour les systèmes sans fil reconfigurables.





# TABLE OF CONTENTS

<b>Acknowledgments</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Table of Contents</b>	<b>xxvii</b>
<b>Abbreviations</b>	<b>xxxv</b>
<b>Introduction</b>	<b>1</b>
<b>1 CONCEPTS AND RELATED WORKS</b>	<b>7</b>
1.1 Introduction . . . . .	7
1.2 Wireless Communications . . . . .	8
1.2.1 Wireless Channels . . . . .	8
1.2.2 OFDM . . . . .	9
1.2.3 Wireless Communication Standards . . . . .	10
1.2.3.1 Wireless Local Area Networks . . . . .	10
1.2.3.2 Wireless Metropolitan Area Networks . . . . .	12
1.2.3.3 Cellular Networks . . . . .	12
1.2.3.4 Internet of Things Networks . . . . .	12
1.3 Adaptive Techniques and Reconfigurable Radio . . . . .	13
1.4 Vertical Handover in Heterogeneous Networks . . . . .	16
1.4.1 Vertical Handover Algorithms . . . . .	16
1.4.2 VHAs Based on Machine Learning . . . . .	19
1.4.2.1 Machine Learning . . . . .	19
1.4.2.2 Neural Networks . . . . .	20
1.4.2.3 Reinforcement Learning . . . . .	20
1.4.2.4 Applying the Machine Learning Technique in the VHA context . . . . .	21

1.4.3	Unified Reconfigurable Communication Chain for Multiple Wireless Standards . . . . .	23
1.5	Flexible Reconfigurable Platforms for Communications Systems . .	25
1.5.1	Field Programmable Gate Array . . . . .	25
1.5.2	Dynamic Partial Reconfiguration (DPR) . . . . .	27
1.5.3	ARM-FPGA Platform . . . . .	28
1.5.4	Embedded Operating Systems in SoC Devices . . . . .	30
1.6	Applying Partial Reconfiguration in Context of Reconfigurable Radios	32
1.7	Summary . . . . .	34
<b>2</b>	<b>ARM-FPGA BASED PLATFORM FOR AUTOMATED ADAPTIVE WI- RELESS COMMUNICATION SYSTEMS USING PARTIAL RECONFIGU- RATION TECHNIQUE</b>	<b>37</b>
2.1	Introduction . . . . .	37
2.2	General System Design . . . . .	38
2.2.1	Overview for ARM-FPGA platform . . . . .	38
2.2.2	Xilinx Vivado Tools . . . . .	39
2.2.3	Processing System Modules- PS part . . . . .	39
2.2.3.1	Micro-Kernel . . . . .	40
2.2.3.2	Configuration Controller . . . . .	40
2.2.3.3	Parameters Provider . . . . .	42
2.2.3.4	Hardware Updater . . . . .	42
2.3	Auto Adaptive OFDM transmitter . . . . .	43
2.3.1	Reconfigurable Hardware . . . . .	44
2.3.2	Adaptive processes . . . . .	47
2.4	Experimental Results . . . . .	50
2.5	Summary . . . . .	54
<b>3</b>	<b>RECONFIGURABLE UNIFIED PHY FOR MULTI WIRELESS SYS- TEMS USING PARTIAL RECONFIGURATION</b>	<b>57</b>
3.1	Introduction . . . . .	57
3.2	ARM-FPGA Platform for Switching between Multi Wireless Sys- tems . . . . .	57
3.2.1	Software Part . . . . .	58
3.2.2	Hardware Part . . . . .	59
3.3	Proposed Reconfigurable Architectures . . . . .	60
3.4	Case Study : WiFi-WiMax Unified Reconfigurable chain . . . . .	63

3.4.1	IEEE 802.11 WiFi . . . . .	64
3.4.2	IEEE 802.16 WiMax . . . . .	64
3.4.3	Unified chain Design using PR . . . . .	64
3.5	Experimental Results . . . . .	66
3.5.1	Performance Evaluation for MRBA and ORBA . . . . .	66
3.5.2	Experimental Results for the Unified Chain for WiFi-WiMax . . . . .	68
3.6	Summary . . . . .	71
<b>4</b>	<b>INTELLIGENT VHA FOR WIRELESS NETWORKS USING NEURAL NETWORKS AND REINFORCEMENT LEARNING</b>	<b>73</b>
4.1	Introduction . . . . .	73
4.2	Heterogeneous Networks and Vertical Handover . . . . .	73
4.3	VHA for WiFi and WiMax Heterogeneous Network . . . . .	74
4.3.1	Vertical Handover Algorithm based on Adaptive Scoring System . . . . .	75
4.4	Proposed VHA based on Machine Learning . . . . .	80
4.4.1	Proposed System Design . . . . .	80
4.4.1.1	Parameters Provider . . . . .	80
4.4.1.2	The Neural Network . . . . .	81
4.4.1.3	Adaptive Scoring System . . . . .	82
4.4.1.4	Learning Agent . . . . .	82
4.4.1.5	Reward Provider . . . . .	83
4.4.1.6	Decision Selector . . . . .	83
4.4.1.7	The Configuration Controller . . . . .	84
4.4.2	NN based VHA Vs. standard VHA . . . . .	84
4.5	Experimental Results . . . . .	85
4.6	Summary . . . . .	91
<b>5</b>	<b>IMPLEMENTATION AND REAL TIME POWER MEASUREMENTS</b>	<b>93</b>
5.1	Introduction . . . . .	93
5.2	Implementation and Power Analysis for Automated Adapted Reconfigurable OFDM Transmitter . . . . .	93
5.2.1	Implementation of the OFDM Transmitter on FPGA . . . . .	94
5.2.2	Applying Partial Reconfiguration on OFDM Transmitter . . . . .	95
5.2.3	Automated Adapted Reconfigurable OFDM Transmitter . . . . .	97
5.2.4	On Chip Power Results . . . . .	98

5.3	Real-Time Measurements and Analysis of Dynamic and PR Power Consumption for Zynq SoC Devices . . . . .	100
5.3.1	BackGround and Related Work . . . . .	100
5.3.2	Experimental Setup . . . . .	101
5.3.3	Experimental Results . . . . .	105
5.3.3.1	PL Power Measurements . . . . .	105
5.3.3.2	PS Power Measurements . . . . .	107
5.4	Summary . . . . .	110
<b>6</b>	<b>Conclusion and Perspectives</b>	<b>111</b>
6.1	Summary . . . . .	111
6.2	Perspectives and Future Works . . . . .	114
	<b>Appendix A DEMONSTRATIONS</b>	<b>115</b>
A.1	Introduction . . . . .	115
A.1.1	Ker-ONE : Embedded Virtualization Approach with Dynamic Reconfigurable Accelerators Management . . . . .	115
A.1.1.1	Demonstrations Description . . . . .	117
A.1.2	WiFi-WiMax Vertical Handover on an ARM-FPGA Platform with Partial Reconfiguration . . . . .	119
A.1.2.1	WiFi-WiMax Handover on ARM-FPGA Platform with PR Technique . . . . .	119
A.1.2.2	Demonstration Description . . . . .	119
A.1.2.3	VHA on ZedBoard with Partial Reconfiguration . . . . .	119
A.1.2.4	End-Node Mobility Simulator on the PC . . . . .	120
A.1.2.5	Connection between the ZedBoard and the Simulator	121
	<b>Appendix B EVALUATION BOARDS</b>	<b>123</b>
B.1	ZedBoard . . . . .	123
B.1.1	FEATURES . . . . .	123
B.1.2	TARGET APPLICATIONS . . . . .	124
B.2	ZYNQ-7000 SOC ZC706 Evaluation Kit . . . . .	124
B.2.1	Key Features and Benefits . . . . .	125
B.3	ZYNQ-7000 SOC ZC702 Evaluation Kit . . . . .	125
B.3.1	Key Features and Benefits . . . . .	126
	<b>List of Figures</b>	<b>129</b>

TABLE OF CONTENTS

xxxiii

**List of Tables**

**133**

**Bibliography**

**133**



# ABBREVIATIONS

**MAC** : Media Access Control **Multi-RAT** :Multiple Radio Access Technologies  
**VHA** : Vertical Handover Algorithm  
**QoS** : Quality Of Service  
**FPGA** : Field Programmable Gate Array  
**PR** : Partial Reconfigurable  
**PS** : Processing System  
**PL** : Programmable Logic  
**OFDM** : Orthogonal Frequency Division Multiplexing  
**NN** : Neural Network  
**IFFT** : Inverse Fast Fourier Transform  
**WLAN** : Wireless Local Area Network  
**WMAN** : Wireless Metropolitan Area Networks  
**IoT** : Internet Of Things  
**Wi-Fi** : Wireless Fidelity  
**GSM** : Global System for Mobile Communications  
**LTE** : Long Term Evolution  
**RSSI** : Received Signal Strength Indicator  
**SNR** : Signal-to-Noise Ratio  
**BER** : Bit Error Rate  
**NN** :Neural Network  
**ASIC** : Application-Specific Integrated Circuit  
**DSP** : Digital Signal Processor  
**DPR** : Dynamic Partial Reconfiguration  
**RM** : Reconfigurable Module  
**HDL** : Hardware Description Language  
**SoC** : System on Chip  
**MC-CDMA** : Multi Carrier Code Division Multiple Access  
**PSK** : Phase Shift Modulation



**QAM** : Quadrature Amplitude Modulation

**SDR** : Software Defined Radio

**ICAP** : Internal Configuration Access Port

**PCAP** : Processor Configuration Access Port

**IP** : Intellectual Property

**DMA** : Direct-Memory Access

**OS** : Operating System

**GPIO** : General-Purpose Input Output

**AXI** : Advanced EXtensible Interface

**GIC** : Generic Interrupt Controller

**SRAM** : Static Random Access Memory

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

# INTRODUCTION

## Context

Wireless communication was defined as the transmission and reception of signals by electromagnetic systems. It was initially used around 1890 in wireless telegraphy. In the 90's it was used mainly for the voice telecommunication which was known as cellular network. Today, researchers keep working on achieving best wireless communication performance in terms of data rate, coverage, and mobility.

In the last decade, wireless communication systems have greatly evolved in terms of mobility, covered range, and throughput. A lot of standards with different specifications, both in the physical layer (PHY) (bandwidth, throughput, and modulation encoding techniques) and media access control (MAC) layer, have been provided to meet different technical requirements and give access to new services. Nowadays, wireless communications systems are everywhere, in our homes, offices, metros, urban, and in space. Furthermore these systems are now used for machine to machine communication in an Internet of Things (IOT) to go beyond the standard Internet.

The wireless channel is one of the most challenging communication channels. Due to the characteristics of this channel, the existence of obstacles in the signal path, and the mobility of the end-nodes, the form of the transmitted signal will not be the same at the receiver side. Since the wireless channel is considered unstable, the wireless system at the end-node may be reconfigurable and some modules in the wireless communication chain could be reconfigured to optimize the usage of the available channel capacity. Also today, users are facing a huge diversity of communication networks and protocols that are generally very complex and heterogeneous. This is especially true for mobile devices that have to deal with multiple radio access technologies (Multi-RAT) during their mobility.

Nowadays the requirements of the applications used at the end-node consist in using the best available wireless standard. Throughput, delay, and other quality

of service parameters are variable due to the mobility of the user. This is effected by the characteristics of the wireless channel used by the wireless standard and the variable number of connected users to the wireless network. To achieve best performance, the end device should switch to the best available wireless standard.

In this context, it seems obvious that one could benefit from using reconfigurable wireless systems. The open questions are how and when reconfigurable radios should be reconfigured? Which parts of the wireless communication chain have to be reconfigured during runtime? What is the overhead of real time reconfiguration? Is it possible to have one reconfigurable chain for multiple wireless standards? How to decide which is the best standard and how to switch the communication through it?

In order to decide when and what to reconfigure in the communication chain, adaptive techniques were proposed to enable applying modifications in real-time to both PHY or MAC layers of a wireless system. Using these techniques, modifications are performed according to specific conditions and to the state of the communication channel within the overall network. This allows achieving the best performance by adapting and reconfiguring the communication system in real-time without interruption. The operation of switching between radio access technologies is denoted as vertical handover. Vertical Handover Algorithms (VHA) have been proposed to enable selecting the best available wireless communication system at the end-node. Based on multiple parameters coming from different layers, many simple and complex VHAs have been proposed to achieve better Quality of Service (QoS) or most power efficient communications. Machine learning algorithms are one of the promising techniques used in such VHAs to provide intelligence to the end-node device.

When a decision is made, the system needs to execute and reconfigure the communication chain. This reconfiguration step can be applied to software as proposed in Software Defined Radio (SDR). On the other hand, hardware reconfiguration of wireless communication systems seems also a promising solution for implementing adaptive techniques and systems dealing with multiple wireless standards. In this case, some system functions may be modified during runtime, while other functions continue to run without any interruption. Field Programmable Gate Arrays (FPGA)s constitute efficient circuits to implement such reconfigurable systems. Such devices are now sufficiently mature to implement very complex systems with a high level of performance. Partial Reconfiguration (PR) is one of the interesting features that has been added by FPGAs' vendors to ensure even more flexibility. PR provides the ability to reconfigure a specific modules on the FPGA during

runtime. Furthermore, partial reconfiguration decreases the time needed to reconfigure parts of the circuit, and reduces its power consumption by considering only partial bit-streams instead of complete ones.

Another interesting feature in recent FPGA devices is the presence of embedded hard processor cores that are implemented with the hardware FPGA fabric, in the same chip. For example, the ZYNQ SoC devices from Xilinx feature a dual-core ARM Cortex-A9 based Processing System (PS) as well as a Programmable Logic (PL) within a single device. Compared to their soft core processors counterparts, these processors offer much more computing power and speed. SoC devices are appropriate for designing joined software and hardware (SW/HW) systems. The existence of such systems in the end-devices allows designing high performance smart communication systems.

In this context, it has become very interesting to consider a multi-standards device, capable of implementing several MAC/PHY layers within the same chip. In general, this is performed by implementing several radio access technologies, in parallel, within the same chip. This solution is obviously very costly and not satisfactory in terms of hardware resources usage. Moreover, only few protocols can be handled in this case.

It is important to understand and study the mechanisms of applying PR on ARM-FPGA platform in the context of reconfigurable wireless system. What are the requirements and needs for creating a reconfigurable wireless system on both software and hardware parts? How do these parts interact with each other and how the reconfiguration is managed? For that, PS modules have been proposed to control the PR operation on PL in context of wireless communication systems. It is also important to understand what should be taken into consideration in the hardware design when applying partial reconfiguration. In addition, some measurements should be done to extract specific parameters in order to evaluate the performance of the system. These parameters include the reconfiguration time, on-chip power consumption and reserved hardware resources. Also, it was interesting to perform real time measurements for PR power consumption.

To study the proposed system and in order to achieve measurements, a reconfigurable Orthogonal Frequency Division Multiplexing (OFDM) transmitter, which is considered as one of the most modulation schemes in current standards, has been implemented. Adaptive and PR techniques have been applied to reconfigure the implemented chain. As a case study, switching between WiFi-WiMax standards using PR has been considered and implemented and an unified chain for both standards was implemented. In addition, VHA based on adaptive scoring system

for both standards was proposed. It was very interesting to benefit from machine learning algorithms such as Neural Network (NN) to improve the decision of the proposed VHA and make it more intelligent.

This thesis proposes to answer the questions by benefiting from the PR technique in the context of reconfigurable wireless communication systems on ARM-FPGA platform. In our research, we propose a general architecture for such type of systems. Our approach has been implemented and tested in specific scenarios. We have also run extensive experiments on our system to evaluate the resources, time, and power consumption of the proposed approach.

## Manuscript organization

This thesis is organized as follows : In Chapter 1, we introduce the general techniques and concepts used throughout this thesis. Also, we present the related works concerning the topics used in the research. In Chapter 2, we propose an ARM-FPGA platform for reconfigurable radios using the partial reconfiguration technique. The proposed system is applied on an auto adapted OFDM transmitter as a case study. The results for power consumption, resources utilization, and reconfiguration time are presented. In Chapter 3, the ARM-FPGA platform is improved to be applied for switching between multiple wireless systems. An unified chain for WiFi-WiMax is tested as a case study to extract the required results. In Chapter 4, we describe a vertical handover algorithm for switching between multiple wireless systems based on a neural network. The performance of our system in terms of real-time power consumption are presented in Chapter 5. We conclude this thesis and give the perspectives of our work in Chapter 6.

## List of scientific contributions

### Journals :

- "Applying Partial Reconfiguration Technique on ARM-FPGA Systems in Context of Vertical Handover in Wireless Heterogeneous Networks", "**Mohamad-Al-Fadl Rihani**, Mohamad Mroue, Jean-Christophe Prévotet, Fabienne Nouvel, and Yasser Mohanna", International Journal of Digital Information and Wireless Communications (IJDIWC), 2018(1), p.70.
- "ARM-FPGA-based platform for reconfigurable wireless communication

systems using partial reconfiguration", "**Mohamad-Al-Fadl Rihani**, Mohamad Mroue, Jean-Christophe Prévotet, Fabienne Nouvel, and Yasser Mohanna", EURASIP Journal on Embedded Systems, 2017(1), p.35.

## Conferences :

- "A Neural Network Based Handover for Multi-RAT Heterogeneous Networks with Learning Agent", "**Mohamad-Al-Fadl Rihani**, Mohamad Mroue, Jean-Christophe Prévotet, Fabienne Nouvel, and Yasser Mohanna", RECOG, LILLE, France, 2018.
- "Demo : WiFi-WiMax Vertical Handover on an ARM-FPGA Platform with Partial Reconfiguration", "**Mohamad-Al-Fadl Rihani**, Jean-Christophe Prévotet, Fabienne Nouvel, Mohamad Mroue, and Yasser Mohanna", DASIP, Dresden Germany, 2017.
- "Vertical Handover Algorithm for WiFi-WiMax Systems Using Partial Reconfiguration Technique", "**Mohamad-Al-Fadl Rihani**, Jean-Christophe Prévotet, Fabienne Nouvel, Mohamad Mroue, and Yasser Mohanna", GDR-SoC-SIP, Bordeaux France, 2017.
- "ARM-FPGA Based Platform for Automated Adaptive Wireless Communication Systems Using Partial Reconfiguration Technique", "**Mohamad-Al-Fadl Rihani**, Jean-Christophe Prévotet, Fabienne Nouvel, Mohamad Mroue, and Yasser Mohanna", DASIP, Rennes France, 2016.
- "Dynamic and Partial Reconfiguration Power Consumption Runtime Measurements Analysis for ZYNQ SoC Devices", "**Mohamad-Al-Fadl Rihani**, Fabienne Nouvel, Jean-Christophe Prévotet, Mohamad Mroue, Jordane Loral, and Yasser Mohanna", ISWCS, Poznan Poland, 2016.
- "Ker-ONE : Embedded Virtualization Approach with Dynamic Reconfigurable Accelerators Management", "Tian Xia, **Mohamad-Al-Fadl Rihani**, Jean-Christophe Prévotet, and Fabienne Nouvel", DASIP, Rennes France, 2016.
- "Power Analysis for Automated Adapted Reconfigurable OFDM Transmitter", "**Mohamad-Al-Fadl Rihani**, Jean-Christophe Prévotet, Fabienne Nouvel, Mohamad Mroue, and Yasser Mohanna", URSI, Rennes France, 2016.



# CHAPTER 1

## CONCEPTS AND RELATED WORKS

### 1.1 Introduction

In this chapter, the main concepts of the ideas covered in this thesis are explained. The state of the art and related works are also presented. First, wireless communication main concepts as well as wireless standards are presented. Next, we discuss reconfigurable radios and the idea of adaptive techniques. Switching between multiple wireless standard is also considered as reconfiguring the wireless communication system. Switching between multiple wireless standards is called vertical handover. This is also presented with the notion of heterogeneous wireless networks and the vertical handover algorithms concepts. Vertical handover algorithms, which decide when to switch from a wireless standard to another, can be based on machine learning techniques. Definition and discussion about machine learning techniques and related works for benefiting from those techniques in context of vertical handover are also presented in this chapter. After discussing the needs of reconfigurable wireless systems in the context of adaptive techniques or in switching between multiple wireless standard it is important to present a flexible reconfigurable platform. The main feature of the reconfigurable platform as well as the flexible reconfiguration technique are defined and presented. The related works dealing with reconfiguration technique in the context of reconfigurable wireless systems are discussed at the end of the chapter.



## 1.2 Wireless Communications

In the last decades, wireless communications have become ubiquitous in our daily lives. Due to the rise of Internet, end-users always need to be connected indoor and outdoor, when they are stationary or mobility. For this reason, it is very important to get wireless networks availability wherever users go. In addition to its coverage, wireless communications networks should also fulfill the demands of the end-users in terms of stable connectivity and QoS. The wireless channel effect the wireless communications due to its characteristics. The characteristics of the wireless channel is described in the next section.

### 1.2.1 Wireless Channels

Wireless channels have been considered as some of the most difficult channels for transmitting and receiving data. Wireless signals that are transmitted from a transmitter to a receiver are generally affected by multiple factors due to the existence of obstacles and buildings and is particularly sensitive to the mobility of end-users. The wireless signal power may be degraded or even lost because of the propagation mechanisms as free space loss, diffraction, reflection, and refraction of the signal. In addition to other challenges like multi-path delay, Doppler effect, and fading, signals should be processed at transmitter and receiver side for not losing information, and to achieve the maximum available bandwidth on the wireless channel [Kor11].

The aim of digital communication systems is to transmit the information with the highest reliability. The general architecture of such a system consists of three basic components : the transmitter, the wireless channel and the receiver as illustrated in FIGURE 1.1. At the transmitter side, the source sends the information to the source encoder. The source encoder converts this signal into a binary data stream. The bits stream is encoded by a channel encoder which outputs a code-word. The channel encoder encodes the data to make them more reliable and have the ability to correct the errors caused by noise or interference through the transmission channel. The coded data are then mapped into a carrier signal through the modulator. Then the signal passes through radio propagation channels where many perturbations may affect the signal. At the receiver side, an altered version of the transmitted signal is received. The reverse operation of signal processing is applied. The demodulator and the decoder recover the original signal and generate an estimate of the transmitted information and send it to the destination (sink).

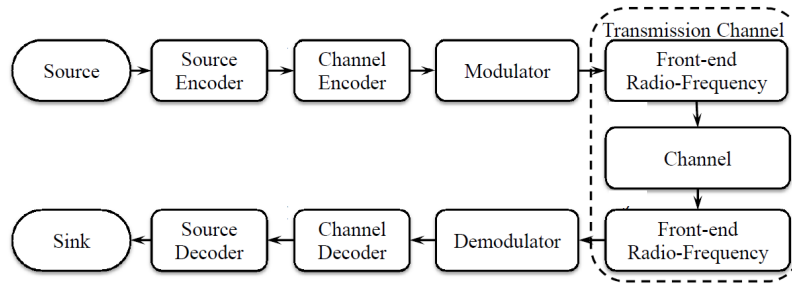


FIGURE 1.1 – Wireless channel effects on transmitted signals

Researchers have proposed multiple modulation and multiplexing techniques to maximize the channel capacity. One of the famous multiplexing technique is the Orthogonal Frequency-Division Multiplexing (OFDM) that will be explained in the next section.

### 1.2.2 OFDM

First introduced in the 1950's, the OFDM is a very popular multiplexing technique allowing a high data rate transmission over wireless mediums [Cha66]. OFDM provides good performance against multi-path interference on large distances. It has become the core of most 4G fixed and mobile wireless communication systems. This technique divides high data streams into multiple low rate data streams as shown in FIGURE 1.2. Then, the information is modulated and transmitted over multiple sub-carriers. This technique was robust against fading channels by minimizing the Interference Inter Symbol (ISI) between sub-carriers. Sub-carriers are orthogonal to each other in order to avoid the interference between them (thanks to the Inverse Fast Fourier Transform (IFFT) function). A cyclic prefix (CP) is added before transmission at the end of the OFDM symbol. The cyclic prefix must be chosen larger than the delay spread of the channel to completely remove the ISI. The OFDM modulation and demodulation are presented in FIGURE 1.3.

The input data stream is modulated by a modulator, resulting in a complex symbol stream  $X[0], X[1], \dots, X[N - 1]$ . The input serial data are arranged in parallel formation according to the size of the IFFT. The data bit-stream is mapped to symbols, I and Q values of symbols are now assigned to sub-carriers. These sub-carriers are transformed from frequency domain to time domain through the IFFT block. The IFFT yields the OFDM symbol consisting of the sequence  $x[n] =$

$x[0], \dots, x[N - 1]$  of length  $N$ , where

$$x[n] = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} X[i] e^{j2\pi ni/N}, 0 \leq n \leq N - 1 \quad (1.1)$$

CP is inserted at the end of the vector and then the time symbols are transmitted serially to wireless channel.

At the receiver, received symbols are arranged in parallel formation then CP is removed. After that, the FFT block transforms symbols from time domain to frequency domain. Finally, The frequency symbols are transferred serially then to the demodulator [Gol05].

OFDM technique is sensitive to Doppler shifting, which leads to a loss of orthogonality between sub-carriers. The high Peak-to-Average Power Ratio (PAPR) is another drawback of OFDM technique. Because of its good performance in fading wireless channels, the OFDM is considered as the core of the PHY of the most popular wireless communication standards which we will mention in the next section.

### 1.2.3 Wireless Communication Standards

Wireless communication systems have greatly evolved in terms of mobility, covered range, and throughput. A lot of standards with different specifications, both in the PHY (bandwidth, throughput, and modulation encoding techniques) and MAC layers, have been provided to meet different technical requirements and give access to new services. The most famous types of wireless networks are the Wireless Local Area Network (WLAN), Wireless Metropolitan Area Networks (WMAN), cellular network and the Internet of things (IoT) networks.

#### 1.2.3.1 Wireless Local Area Networks

WLAN is a type of network that connects several equipments using the wireless media and provides Internet accessibility within a bounded region such as a company, university, house, hospital, etc. Current WLANs are based on IEEE 802.11 standards and are advertised under the wireless fidelity (Wi-Fi) trademark. The IEEE 802.11 standard defines multiple releases, each of a set of MAC and PHY specifications for a Wi-Fi network. Most PHY layers of the WiFi releases are based on the OFDM technique [IEE99]. Designed for indoor, WiFi is considered as short range with high data rate wireless communication standard. Nowadays,

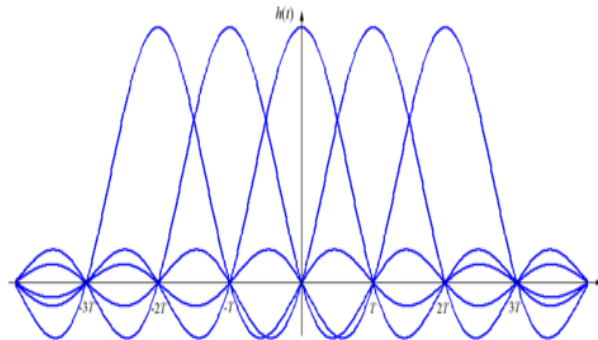


FIGURE 1.2 – OFDM symbol as shown in frequency domain

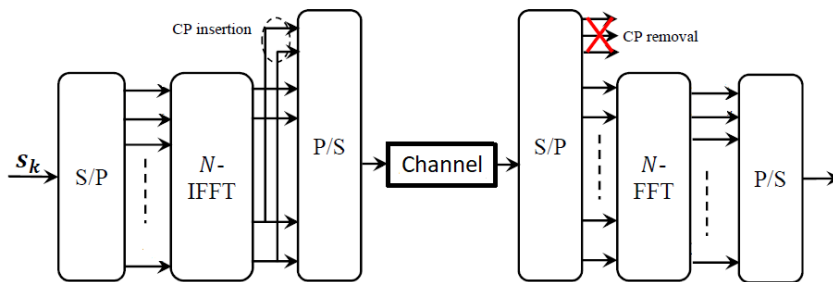


FIGURE 1.3 – OFDM modulation and demodulation process

the performance of WiFi networks is remarkable ; the performance of all supported services is excellent. One main advantage for WLAN networks is the low cost at the operational and maintenance level. When compared to cellular networks, the infrastructure equipment cost is much lower. Therefore, WLAN networks are considered as a low cost competitive radio access technology in low-mobility hot-spot areas. An important disadvantage of WLAN networks is that the throughput at the end node decreases when the distance between a node and an AP increases. Therefore, WLAN networks deployment should be carefully designed, in order to maintain uniform throughput in all the area covered by WLAN APs.

### **1.2.3.2 Wireless Metropolitan Area Networks**

WMAN have been designed to provide Internet accessibility for the users in a larger area than that defined for the WLAN. One of the most popular WMAN standards is the IEEE 802.16 known as WiMax [wim]. WiMax implements full packet switching (IP based) networks and provides wider coverage than WiFi networks. The core of the PHY layer of WiMax is also based on the OFDM technique [Kor11]. The WMANs are designed to provide wider coverage with a data rate and performance similar to the WLAN networks.

### **1.2.3.3 Cellular Networks**

Cellular networks were proposed to cover cities and provide voice and text services for users. Currently, the global system for mobile Communications (GSM) is the most widely-spread cellular system that supports both circuit-switched and packet-switched services. Cellular networks have evolved by providing data services. One of the most famous recent generation of the cellular network is the Long Term Evolution(LTE) [lte] which is considered as 4G. Now researchers are working on 5G which should provide more coverage and throughput. The main goal of cellular networks is to cover wide area while supporting high user mobility. For this reason, these networks are designed in a way that ensures good service for users experiencing high mobility. The importance of cellular networks is mainly characterized by the large geographical coverage region and the huge number of users served by these networks.

Usually the number of connected nodes to the cellular networks is huge, for that cellular networks support multiple hierarchical structures. For example, a single cell could be adjusted to operate in a hot spot area, rural, urban, and suburban areas, by using what we call pico, micro, and macro cells. As the number of connected nodes increases, the data rate for each node decreases also. In some cases, the radio resources are all reserved and the data rate supplied for each user could be extremely low. Therefore, cellular networks could cooperate with WLANs and WMANs to supply better services for users.

### **1.2.3.4 Internet of Things Networks**

IoT is a network that enables communication and data exchange between electronic devices, vehicles, home appliances and any other objects. The IoT network allows to sense or remotely control nodes connected to the network. Such type of

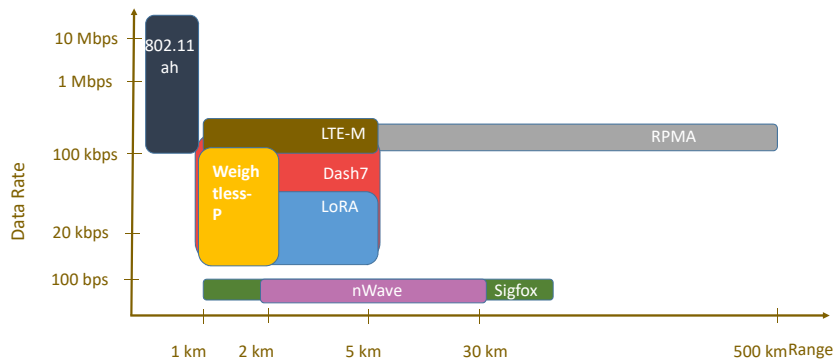


FIGURE 1.4 – Data rate vs. range for IoT connectivity technologies

applications requires long range and low power wireless communications. Many wireless standards are proposed to fulfill the demands of IoT like LoRa [All15], SigFox, NB-IoT, LTE-M [RMGV14], etc ... In FIGURE 1.4 different IoT standards are compared in terms of achieved range and data rate.

We can notice the existence of multiple wireless standards. This will contribute to form heterogeneous networks. End-nodes will need to know how to select the best available wireless standard according to the required preferences.

Before describing the techniques and algorithms that are used to select the best wireless standard and how to reconfigure the wireless communication chain, we first present the methodology of adapting the wireless radio to the channel situation.

### 1.3 Adaptive Techniques and Reconfigurable Radio

Increasing the performance of wireless communication systems has always been one of the first objectives of designers. Due to the characteristics of wireless channels (path loss, fading, shadowing ...), the multiplicity of standards, the frequency allocations, and the mobility features provided by wireless devices, the operating environment has become more and more complex to understand. In FIGURE 1.5, the general four steps (Sense, analyze, decide, and reconfigure) applied by reconfigurable radio are presented [MM99]. This concept aims to create a smart radio which has the ability to reconfigure itself to adapt according to the new wireless

channel status. The first main step is to sense the environment continuously, and then analyze and understand the new wireless channel status. Upon the analysis, the system should decide whether to reconfigure itself or not. If the decision was to reconfigure, reconfiguration is applied and the system starts sensing the environment again. In this case, it is necessary have multiple versions of the wireless radio that are compatible with different status of the wireless channel. The system has the ability to identify the wireless channel status. This allows deciding whether it is necessary to reconfigure the system to another version or not. Each of the four mentioned steps raises new research issues. For example what parameters must be sensed and analyzed to detect the new wireless channel state? Also, what is the optimal decision that can be made or how to apply fast and efficient reconfiguration?

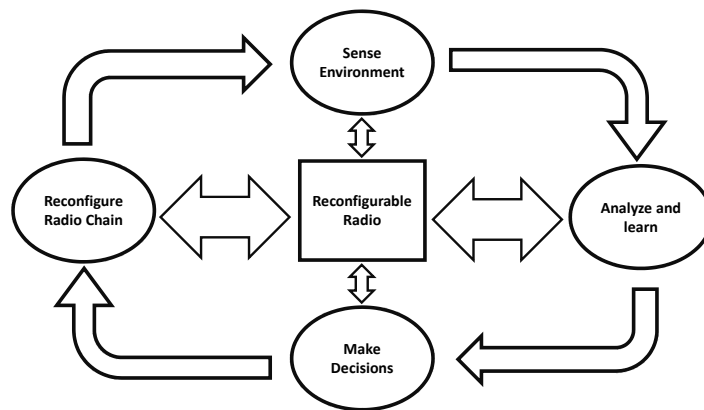


FIGURE 1.5 – Reconfigurable radio block diagram

In this context, researchers have proposed adaptive mechanisms to enable wireless systems to adapt waveforms according to channel properties. Such systems sense, learn, analyze, and decide to reconfigure part of the wireless system. Various low-level parameters such as the Received Signal Strength Indicator (RSSI), Signal-to-Noise Ratio (SNR), Bit Error Rate (BER) have been used to guarantee adaptability. Other high-level parameters, such as resource allocation, QoS, and power consumption, have also been investigated to decide and appropriately adapt the system to the environment in real time. One of the main goals of the adaptive techniques consists in maximizing the channel capacity while minimizing the po-

power consumption. In some cases the situation is based on a trade-off. For example, sometimes adaptive techniques may reconfigure the system to reduce the data rate in order to decrease the power consumption. In other cases it increases the power consumption to increase the data rate.

Adaptive modulation is one of the transmission techniques that has been proposed to set up the best constellation scheme according to the channel conditions. The constellation mapping scheme or the modulation technique is reconfigured with different versions according to specific values of the parameters that are sensed from the wireless channel. Some works in this context are described in [FS], [KH00], and [MN15]. For example, in FIGURE 1.6 from [FS], it is clearly shown how the adaptive modulation provides the best performance compared to static modulation when SNR values are changing. The adaptive modulation decreases the constellation mapping rate as the SNR decreases which will decrease the bit error rate. On the other hand when the SNR increases, a modulator with higher data rate is selected to increase the throughput.

Adaptive channel encoding and code rate is another adaptive technique proposed to select the best encoder or the code rate according to the channel status. The encoder has the role to add coded bits to data in order to provide reliable communication and to allow the correction of some errors caused by the channel. Some types of encoders add more coded bits than others but with a better error correction rate. This trade-off between decreasing the data rate and BER can be adapted to obtain an optimal solution using adaptive encoding techniques. Some works in this context are presented in [SPA10], [LTG04], and [DF03].

Other algorithms and methodologies have also been proposed to deal with bandwidth adaptation. In this case, the system is able to adapt the bandwidth size according to the spectrum status. For example, in OFDM systems, FFT and sub-carrier mapper blocks can be adapted according to parameters provided by the device and by the network as presented in [ZKS07] and [FBF16]. In [ZEA05] and [LWN09], the authors show that it is important to adapt the FFT length with respect to the power consumption. The results presented in these two papers describe how the power consumption increases as the size of the FFT increases.

Applying adaptive techniques on wireless communications requires implementing an algorithm that senses the environment and accordingly reconfigures specific blocks in the PHY layer during runtime for example the encoder, the modulator, and IFFT blocks.

In this section, we have defined and mentioned some of the adaptive techniques used to improve the quality of the wireless communication and make an efficient



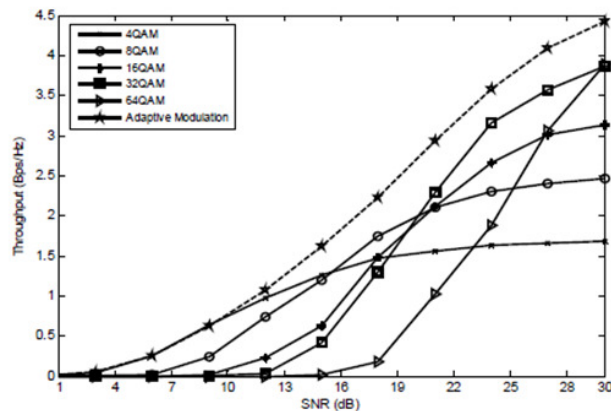


FIGURE 1.6 – Adaptive modulation performance results

decision to reconfigure the wireless chain. Also, there are other types of techniques and algorithms to select the best available wireless standard and decide to reconfigure the communication chain to transmit data through the selected standard. These are presented in the next section.

## 1.4 Vertical Handover in Heterogeneous Networks

The operation of switching between two standards is called vertical handover. The VHAs algorithms decide when to apply the handover. The execution of the handover may be performed by reconfiguring a wireless chain to switch from a standard to another.

### 1.4.1 Vertical Handover Algorithms

In heterogeneous wireless networks, the concept of being always connected has progressively move towards being always best connected. That is, not only being always connected, but also being connected to the best available network at any time.

Many Vertical Handover Algorithms (VHA) have been developed by researchers to decide which wireless standard in heterogeneous networks is the best for communication. VHAs read and study multiple parameters from the wireless chan-

nel, network, user application, and devices to take such a decision. An example of vertical handover in WiFi and WiMax heterogeneous networks is shown in FIGURE 1.7.

The key of taking a good decision is to collect and study as much parameters as possible, from the environment. These parameters can be divided into four categories :

1. Wireless Channel : Reading parameters from wireless channel inform the VHA about the status of the channel. We can mention some used parameters : RSS, SNR, and BER.
2. Network Information : These parameters inform the VHA about the status of the network. The important parameters that provide information about the network : number of connected nodes to the network, data rate, delay, jitter, and cost.
3. User Parameters : These parameters reflect user data requirements at the application layer ex : data rate, and high priority data.
4. System Parameters : Such type of parameters inform the algorithm about the status of the device such as : power consumption, battery level, and device speed.

Most of the proposed VHAs can be divided into three main steps :

- Trigger : While sensing the environment, a main change is detected which triggers the VHA to process the parameters.
- Decision : In this step, the VHA will read the values of the parameters and decide to keep on operating with the current wireless standard or to switch to another available wireless standard.
- Execution : After deciding to switch to another wireless standard, the data flow should be directed on the device to the MAC/PHY of the selected standard. In addition, signaling between access points should be performed to provide seamless handover.

In [ABG14], the authors provide a detailed state of the art of the existing vertical handover algorithms. The survey defines five categories of VHAs :

1. RSS based Schemes : In RSS based decision algorithms, RSS is used to decide which available network is better.
2. QoS based Schemes : Such VHA must achieve best QoS, and make optimal handover decision using available bandwidth, SINR, and user preferences parameters.

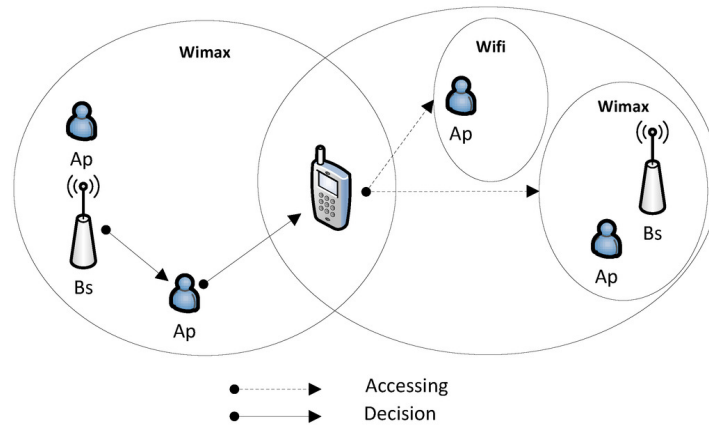


FIGURE 1.7 – Vertical Handover in WiFi-WiMax Heterogeneous Network

3. Decision Function based Schemes : Such vertical handover algorithms generalize the cost, utility, score and policy based functions.
4. Intelligence based Schemes : These schemes require some tools for correlating, analyzing, and reporting the network using some parameters.
5. Context based Schemes : In this category, decisions are made, based on the the information shared between users and access points.

In the context of vertical handover in heterogeneous networks, several works can be mentioned. In [KJU<sup>+</sup>15], the authors proposed an algorithm composed of two steps. In the first step, the handover is triggered according to the data rate requested by the user. Before starting the handover, the system evaluates the speed of the node. If the speed is higher than a certain predefined threshold that does not allow benefiting efficiently from switching to another system, the handover is canceled and considered as unnecessary. Otherwise, if the speed is appropriate, the available networks are compared according to five parameters : the bandwidth, jitter, delay, cost and bit error. Then, the best standard is selected and the vertical handover is performed if necessary.

In [JH09], the authors propose an adaptive network scanning algorithm in the context of heterogeneous wireless networks. In this work, the main idea is to avoid unnecessary scanning processes. As a result the power consumption is reduced. The device speed is the main parameter that the algorithm considers to adapt the scanning period. A speed limit is defined for each wireless standard.

In [ARS10], the authors propose a vertical handover algorithm between LTE and WLAN consisting of 3 steps. The first step aims to initiate the handover based on RSS. In the second step, a fuzzy logic system compares four parameters provided by the two networks (Bandwidth, SNR, Battery Life, and Network Load) to decide which network provides the best connectivity. In the third step, a seamless vertical handover is executed.

Based on the SINR, the authors in [AMM12] propose a vertical handover between WiFi and WiMax. According to a custom mathematical function, the equivalent SINR value for each network is computed and compared. The handover decision is based on the best equivalent SINR value.

VHAs may be more intelligent if based on machine learning algorithms. In the next section, machine learning techniques are defined and VHAs based on ML are mentioned.

## 1.4.2 VHAs Based on Machine Learning

The most promising proposed VHAs are those based on the machine learning techniques. They can be categorized as intelligence based schemes defined in [ABG14] with the other fuzzy logic based VHAs. VHAs usually detect available wireless standards and make a decision based on the current values of certain parameters. On the other hand, VHAs based on machine learning techniques have the ability to study and learn from the history to create an efficient decision.

### 1.4.2.1 Machine Learning

Machine Learning is the science of getting computers to act without being explicitly programmed. It uses algorithms to build analytical models, helping computers to learn from data. It can now be applied to huge quantities of data to create exciting new applications.

Many Machine Learning algorithms can be classified according to two main categories [Chr16] :

- Supervised Learning : In this category, the program is generated by training data which contain samples of inputs and desired outputs. A supervised learning algorithm analyzes the training data and produces a function, which can be used for mapping new input data.
- Unsupervised Learning : This type of learning algorithm finds its own structure based on the history of the inputs and the rewards of the outputs. After

presenting a specific amount of inputs/outputs pairs, it will be able to generate appropriate outputs.

Machine Learning algorithms can also be classified according to the following categories :

- Classification : This category is based on algorithms where inputs are divided into two or more classes.
- Regression : In this category, outputs are continuous and not discrete.
- Clustering : It consists of unsupervised techniques where the set of inputs is to be divided into groups.

Machine Learning have been successfully applied in different domains such as computers network, economics, industry, marketing, robotics, medical diagnosis, speech and handwriting recognition.

#### 1.4.2.2 Neural Networks

Neural networks (NN) are computing systems inspired by the biological neural networks that constitute the brain. Some of these systems are considered as supervised machine learning algorithm. In this case, NN learns from a set of data to generate outputs.

An NN is based on a collection of connected units or nodes called neurons. Each connection between neurons can transmit a signal from one to another. In NN implementations, the signal at a connection between artificial neurons is a real number, and the output of each neuron is calculated by applying non-linear function of the a weighted of its inputs. Neurons and connections typically have weights that are adjusted as learning proceeds. A particular weight value may increase or decrease the signal at a connection. The NN system is formed of inputs, hidden and output layers as shown in FIGURE 1.8 [Chr16].

NNs have been used in various applications, including computer vision, speech recognition, machine translation, social network filtering, playing board, video games and medical diagnosis. In communications, NNs are also used to make the networks and end-nodes more intelligent.

#### 1.4.2.3 Reinforcement Learning

Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment in order to maximize the reward. It is similar to dynamic programming, in which rewards are recorded after every action and the function is reconfigured to maximize the reward. The

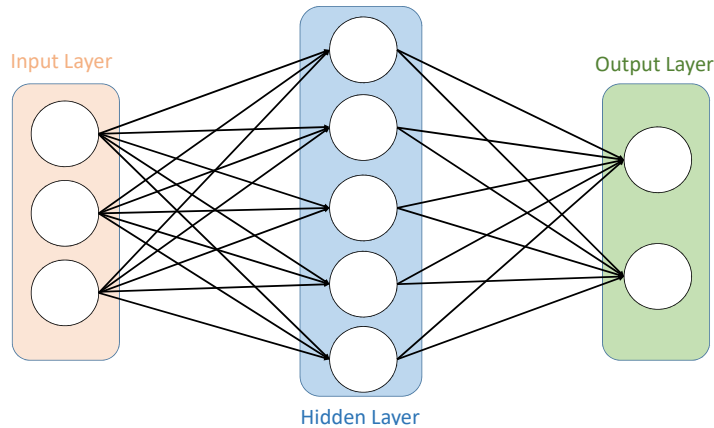


FIGURE 1.8 – Neural network block diagram

algorithm of reinforcement learning is described in FIGURE 1.9. In some cases it is categorized as un-supervised machine learning, whereas in other cases it is considered as an independent category of machine learning. We can mention the Q-learning as one of the famous reinforcement techniques in machine learning. For example, Q-learning can be used in robotics in a scenario where the robot can learn to walk by itself. In this case only the reinforcement algorithm is implemented on the robot. The robot will start changing the speed of the motor in its leg, which can be considered as the actions that transfer from a state to another. After each action the reward will be the stabilization of the robot, so if it falls down then it is a bad reward otherwise it is good. The robot should try different actions until it reaches the best sequence of actions which will provide stable walking. After a while, the robot will learn walking by itself using the reinforcement learning technique.

#### 1.4.2.4 Applying the Machine Learning Technique in the VHA context

Vertical handover in wireless networks has led to a lot of works that benefit from the machine learning techniques. Generally, authors make use of different machine learning algorithms to make a decision and switch from a wireless network to another. This decision is often based on the prediction and the probabilistic study of the collected data history.

In [AA15], the authors use NNs to select the best available wireless standard

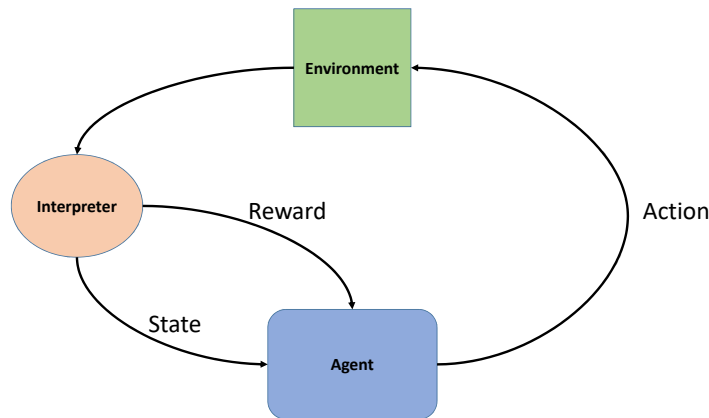


FIGURE 1.9 – Reinforcement learning block diagram

in a WiFi-WiMax-LTE heterogeneous network. In this paper, a vertical handover manager, an attributes Collector, and a neural network system are proposed. The NN has the task of selecting the best standard according to users' preferences. In [YZL11], the authors aim to eliminate failed VHAs by proposing a linear prediction based algorithm to predict the time of vertical handover using location and speed.

The authors in [XLSSL14] proposed a handover between multiple access points of the same standard to communicate through the best available AP. The system is proposed for Vehicular Wireless Networks. Q-Learning was used to benefit from RSS, speed and location to create switching decisions. FIGURE 1.10 illustrates the proposed system by [ZAT15] for mobile heterogeneous networks. This paper combines a fuzzy system, multiple level neural networks and reinforcement learning to switch between multiple wireless systems. The fuzzy logic is responsible for the VH decision. The NN is responsible for learning the Fuzzy Logic Controller (FLC) parameters from the quality indicators. Using Q-learning, the NN system will predict the quality indicator according to the given parameters. RSS, network load are used in this work to determine if handover is acceptable or not.

In [NADS16], a linear regression prediction algorithm is proposed to change over between WiFi access points on a campus. In [JL15], the authors propose a vertical handover between LTE, WiFi, and WIMAX based on Bayesian decision. In [PA10], a hidden Markov chain is used to predict mobility of the node to the

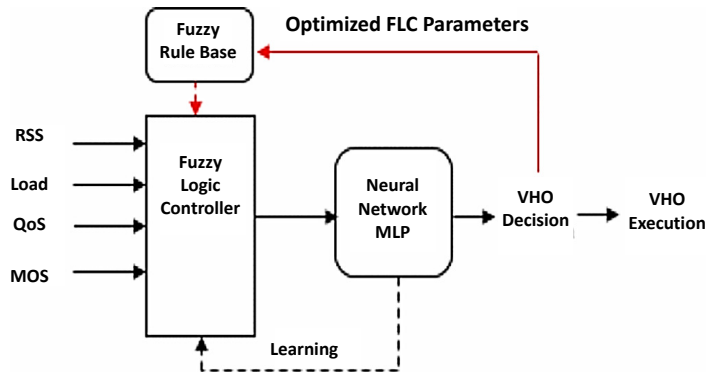


FIGURE 1.10 – General proposed Vertical Handover (VHO) with R-MLP and fuzzy logic controller

AP in order to allocate resources according to the prediction. A lot of other works has also benefited from machine learning techniques, we can mention also some works based on NN in [Bha07], [KM09], [NGS07], and [LWL08].

As a result, VHAs based on machine learning algorithms provide good performance and they have the ability to learn from old records to create prediction functions.

### 1.4.3 Unified Reconfigurable Communication Chain for Multiple Wireless Standards

One of the main steps of vertical handover is the execution step. At the end-node, after deciding to switch from a wireless standard to another, the communication should switch from the current MAC/PHY to the target one. There are two ways of implementing reconfigurable wireless systems. In the standard case, multiple wireless PHY layers are implemented in parallel. On switching, the data flow is re-directed from the blocks of the current standard to the blocks of the other one. Another case is when only one reconfigurable PHY layer for multiple wireless standards is implemented. The blocks of the unified PHY chain are reconfigured when switching from a wireless standard to another. Then communication can switch to the reconfigured standard. An example that is used to switch between two wireless



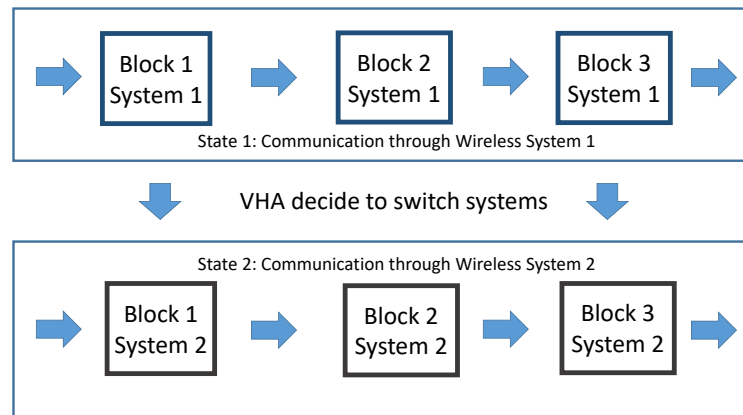


FIGURE 1.11 – An example of switching the physical layer between two wireless systems

systems in PHY layer is shown in FIGURE 1.11. This methodology may face some constraints such as implementing reconfigurable design for wireless standards with different PHY layer architecture and the effect of needed reconfiguration time. So, data from upper layer should be stored until reconfiguration is ended. Also a generic interface for multiple wireless standards should be implemented.

For example, the PHY layers in the popular IEEE standards WiFi and WiMax are based on similar blocks. For that, it was interesting to work on creating a reconfigurable unified chain. In this context, interesting studies for WiMax and WiFi systems are proposed in [NVD<sup>+</sup>07], [DP12], and [RRMS11]. The similarities and differences between WiMax and WiFi physical layers are shown in these articles. These studies describe how WiMax and WiFi physical layers are quite similar in their global architecture with only few differences in specific modules. In the mentioned works, it is explained how it is possible to create a reconfigurable unified chain for both wireless standards.

In this section we have talked about vertical handover between multiple wireless standards. It is noticed that the wireless communication chain can be reconfigured to switch from a standard to another on the execution phase of the VHA. Also, in the previous section, the reason was given why the adaptive techniques require reconfigurable wireless chain. At this step, we can ask the following question : what is the solution to perform reconfiguration for the wireless system and which

digital platform is appropriate to implement a unified chain for multiple wireless standards? This is the topic of the next section.

## 1.5 Flexible Reconfigurable Platforms for Communications Systems

Today, FPGAs are considered as a valuable alternative to their Application Specific Integrated Circuits (ASIC)s counterparts. They provide more flexibility, large bandwidth and are appropriate for signal processing applications. FPGAs are also more advantageous to implement wireless communication systems than Digital Signal Processors (DSP)s due to their intrinsic parallel hardware architecture.

Dynamic Partial Reconfiguration (DPR) is a technique that enables reconfiguring a specific block while the remaining blocks of the system in the FPGA are still running. Furthermore, a promising architecture was proposed by FPGA vendors that now integrates ARM processor cores within their FPGAs, in the same chip.

### 1.5.1 Field Programmable Gate Array

Unlike ASICs, an FPGA is an integrated circuit designed to be configured by a designer after manufacturing. Designs implemented in FPGAs are generally described using a hardware description language (HDL) such as VHDL or Verilog.

FPGA devices contain an array of programmable logic blocks and a hierarchy of reconfigurable interconnects that allow the blocks to be inter-connected together. They consist of a parallel bit-level programming platform. Within FPGAs, logic blocks can be configured to perform complex combinational and sequential functions. In most FPGAs, logic blocks also include memory elements, which may be simple flip-flops or more complex blocks of memory like Random Access Memory (RAM)s and Look Up Tables (LUT)s. Some FPGAs also contain Digital Signal Processing (DSP) blocks to perform complex computations. FPGAs also feature Input/Output (I/O) blocks to buffer the input and output signals received and transmitted from the external pins. In FIGURE 1.13, the internal architecture of an FPGA is presented. From this figure, it is shown how the logic array is divided into Configurable Logic Block (CLB) and how the logic, memory, and DSP blocks are located in the device.

Currently, the most popular FPGA vendors are Xilinx [xil] and Altera [alt]

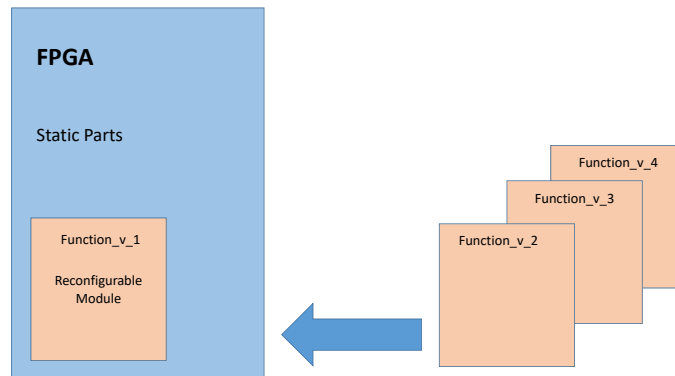


FIGURE 1.12 – Partial reconfiguration on FPGA example

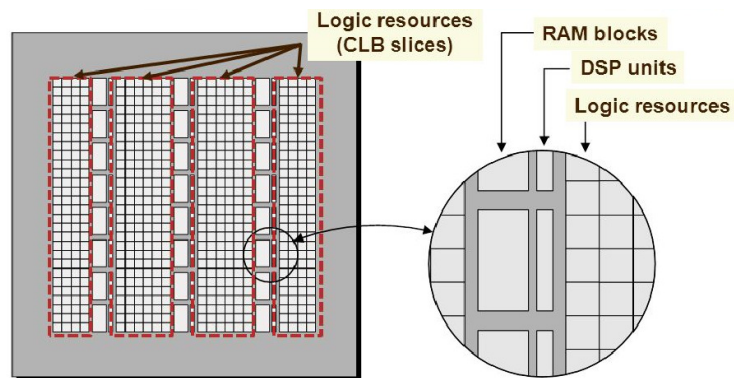


FIGURE 1.13 – FPGA internal architecture

(now Intel). Both vendors produce different versions of FPGAs with different logic gates size, speed, and power consumption. The two vendors provide development software tools like Vivado tools from Xilinx and Quartus from Altera. Using these tools the developer can design both hardware and software systems for SoC devices. The developer can also implement his design and configure the SoC chips of the FPGAs.

### 1.5.2 Dynamic Partial Reconfiguration (DPR)

Partial Reconfiguration is an interesting feature that has been made available in some FPGA families. It provides the ability to reconfigure a subset of logic in the FPGA system. The main idea of DPR is to reconfigure specific blocks within the system while the remaining blocks keep on operating without any interruption. The blocks of the system that can be reconfigured during run-time are called Reconfigurable Modules (RM) whereas the unconcerned blocks are called static modules (FIGURE 1.12).

In addition to providing more flexibility, one of the benefits of partial reconfiguration is the cost reduction that is obtained by reserving a specific area within the FPGA to implement different/multiple functions. It allows reducing resources used on the same FPGA since only one block with multiple functions is implemented rather than having multiple blocks running concurrently. Another benefit of DPR is reduction of in power consumption. This is achieved by reconfiguring small blocks of the system instead of reconfiguring the entire device.

The main disadvantage of using DPR is the reconfiguration time overhead. In Xilinx FPGAs, two main ports are used to partially reconfigure the FPGA during runtime. These are the Internal Configuration Access Port (ICAP) and the Processor Configuration Access Port (PCAP). ICAP is used to reconfigure the FPGA partially by an Intellectual Property (IP) block implemented internally on the FPGA. On the other hand PCAP is used to reconfigure the FPGA from an external processor. The partial bit-stream in both cases is located in the memory and is transferred during PR using a Direct Access Memory (DMA) component. This provides quite similar reconfiguration speeds for both interfaces. The main difference is that ICAP needs additional hardware resources on the FPGA whereas PCAP does not.

In FIGURE 1.14, the transfer of the partial bit-streams from the Secure Digital (SD) to Double Data Rate (DDR) memory, and then to the FPGA through the PCAP is clearly shown. FIGURE 1.15 shows how the internal Microblaze processor

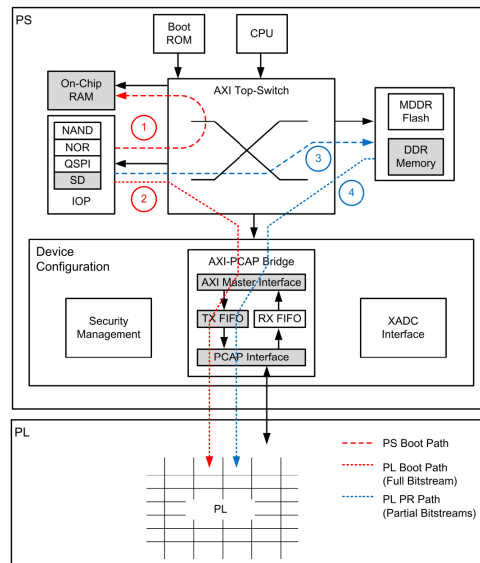


FIGURE 1.14 – Partial reconfiguration flow through PCAP interface

partially reconfigures the RM through the ICAP block. The partial reconfiguration operation is controlled by a process on the Microblaze. The Microblaze processor and ICAP modules are located on the FPGA while the PCAP is located on the SoC between the ARM and the FPGA. The ICAP reconfigures the RM on the FPGA by transferring the bit-stream located in the memory.

One of the important advantages of the DPR is that the different versions of the RMs are located in the memory, which provides the possibility to update or add new versions of a specific RM from a server or cloud on the network.

### 1.5.3 ARM-FPGA Platform

In the last decade, the research on designing embedded microprocessors and FPGA hybrid architectures has become a hot topic. The idea consists in combining the traditional FPGA device and a microprocessor with related peripherals in order to build a complete Reconfigurable SoC (RSoC) device.

As an example, the Zynq 7000 proposed by Xilinx is an SoC device whose architecture consists of a dual-core ARM Cortex A9 processor and an Artix-7 or

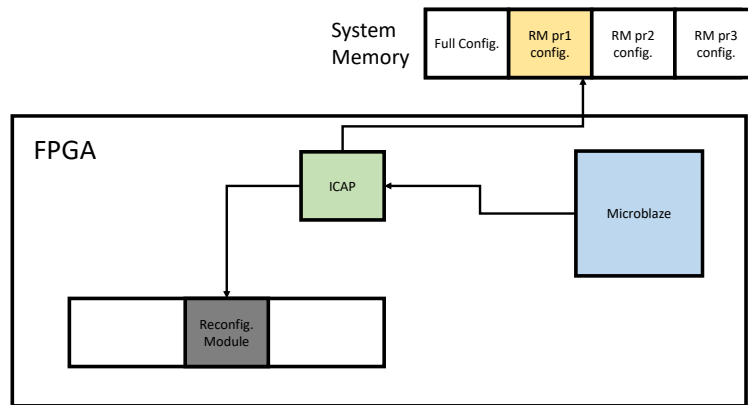


FIGURE 1.15 – Partial reconfiguration flow through ICAP interface and Microblaze

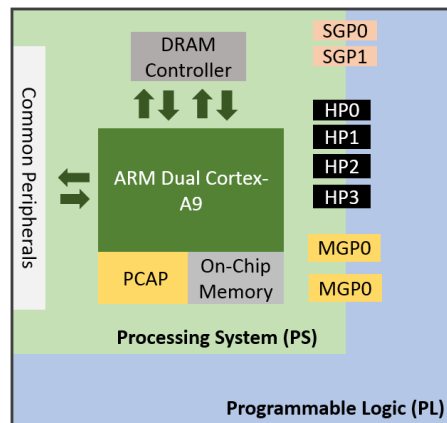


FIGURE 1.16 – A simplified architecture of the Zynq SoC

Kintex-7 FPGA fabric [CEES14].

The Zynq-7000 SoC is divided into two main parts : PS and PL parts. A high speed connection is achieved between the two parts using the Advanced Extensible Interface (AXI) interconnection. A simplified model of the SoC architecture is shown in FIGURE 1.16. The PL part which is equivalent to a classic FPGA fabric is ideal for implementing high speed logic arithmetic and data flow subsystems. On the other hand, the PS part supports software and operating systems. The different AXI buses are shown in the figure, the High Performance (HP) bus and General Purpose (GP) master and slave buses are used to connect both PL and PS parts. As a result, users can benefit from SoC designs to implement Hardware/Software systems where the hardware design is implemented on the PL and software runs in PS with a high speed connection between both parts.

Many development kits make use of the Zynq-7000 SoC. The Zedboard (see FIGURE 1.17) contains a Zynq device with additional peripherals like multiple display interfaces, Ethernet, SD card, DDR3 random access memory, USB-UART, and PS/PL Input/Output expansions. The board contains everything that is necessary to run Operating Systems (OS) like Linux, Windows, Android, or other RTOS-based design. Also, users can develop bare-metal applications or operating systems to manage the creation and usage of the hardware accelerators on the FPGA and benefit from the partial reconfiguration technique.

#### 1.5.4 Embedded Operating Systems in SoC Devices

Operating systems for embedded computer systems are typically designed to be resource and power efficient. The embedded OS are considered to be reliable and are generally compatible with real-time constraints. Most functions found in classical OS are not found in the embedded OS, because the embedded systems are limited in resources and mostly designed for a specific task. Kernels are considered as the core of the computer's OS, it is loaded at initial state and handle input/output requests from pieces of software. For the embedded OS, lightweight micro-kernels have been proposed. Some embedded OS can be mentioned like ucos [uco], RTOS, ...

The basic micro-kernel concept was proposed in [Han70] as a reduced nucleus with fundamental kernel mechanisms. Actual system services were supposed to be implemented at the user-level. Micro-kernel systems follow strictly the principle of least privilege by providing the minimal set of abstractions. The main idea behind micro-kernels design is the low kernel complexity, which results in better feasibility

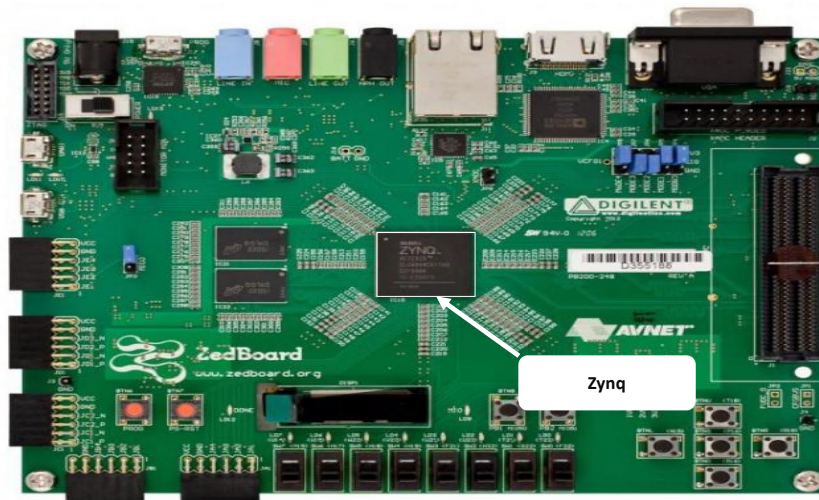


FIGURE 1.17 – ZedBoard with the Zynq-7000 SoC integrated with other peripherals

and security. On ARM-FPGA platform, multiple micro-kernels have been proposed to benefit from this hybrid architecture. Micro-kernels run on the PS part and use blocks on FPGA as hardware accelerators.

As a conclusion, FPGAs provide high performance and ability to apply flexible, dynamic, and partial reconfiguration to implement a system using the DPR technique. With such advantages, FPGA with DPR is considered as a good technique for implementing reconfigurable radios. ARM-FPGA SoC systems provide the ability to create SW/HW applications while benefiting from running multiple processes on the ARM and reconfiguring the hardware modules using partial reconfiguration technique. In the next section we will present some works that have used the PR technique in wireless communication applications.



## 1.6 Applying Partial Reconfiguration in Context of Reconfigurable Radios

Reconfigurable radio systems demand high performance platforms with flexible reconfigurable techniques. Due to its properties, it seems very interesting to benefit from the partial reconfiguration technique of FPGA in wireless communications and reconfigurable radios, since it provides the flexibility to reconfigure hardware modules at runtime. In this section we presents related works used this technique in context of wireless communications.

Numerous works proposed for reconfigurable wireless communication systems and cognitive radio have already taken advantage of the partial reconfiguration technique in FPGAs. In [NR15], a review of partial reconfiguration techniques in wireless communication systems is presented. The authors also suggest benefiting from the partial reconfiguration concept to design and implement a Multi Carrier Code Division Multiple Access (MC-CDMA) system which combines both OFDM and CDMA without presenting the technical details of the proposed system. The authors present the results of the referenced works and discuss the advantages of using partial reconfiguration technique in such systems.

In [Kum12] and [Kum13a], the authors propose an M-Phase Shift Modulation (M-PSK), and M-Quadrature Amplitude Modulation (M-QAM) modems implemented on FPGA with reconfigurable modulation and demodulation modules based on the partial reconfiguration technique for Software Defined Radio (SDR) and Cognitive Radio (CR) applications. In [Kum12], the author uses a Virtex 6 FPGA to implement a PR design for M-PSK modulation and demodulation. The modulator and demodulator are partially reconfigured with different versions (BPSK, QPSK, 8-PSK, and 16-PSK) during runtime by reconfiguring a control register in the FPGA. In [Kum13a], the author describes the design and implementation of 4/8/16/32/64 QAM modulations using the PR technique. Similar to M-PSK, M-QAM RM are partially reconfigured during runtime with multiple versions by configuring a register in the FPGA. When the register is reconfigured, the partial bit-stream is transferred to the RM. In these works, the author compares the PR design with the parallel design and present results that show reduction in reconfiguration time and power consumption values.

Also, in [Kum13b], the author continued his work by proposing a partially reconfigurable OFDM transmitter. The author describes the design and implementation of OFDM transmitters with multiple modulation, and channel coding

schemes. PR technique allows user to swap between multiple OFDM transmitters that are based on 1/2 BPSK, 1/2 QPSK, 3/4 QPSK, 1/2 QAM and 3/4 QAM. This is performed during runtime by configuring a control register in FPGA. The author uses the MicroBlaze soft processor implemented on FPGA to manage the PR process.

In [Kum14], the authors propose a full reconfigurable OFDM transmitter, in which most of the blocks (modulator, encoder, and FFT) of the OFDM are partially reconfigured using the PR technique. However, they do not describe how the partial reconfiguration is managed and controlled in real-time, and under what conditions the system decides to apply the PR process.

In [SABN14], a similar work based on the DPR concept is discussed. It consists in applying DPR on the modulation and encoder blocks of the IEEE 802.11g physical layer. The OFDM physical layer is implemented with various encoding and modulation schemes to achieve different data rates using the PR technique. The design has been implemented in a Xilinx Virtex-5 board. Unfortunately, in this work the authors do not describe how they manage the partial reconfiguration process as well as the required real-time control.

In [VSR<sup>+</sup>13], the authors describe an adaptive reconfigurable transmitter for OFDM-based cognitive radio. In this transmitter, the modulation block is adapted according to the SNR value. The authors implement a simple configuration controller but do not describe the bit-stream transfer to the FPGA. The configuration controller is based on a look up table and it is specified for adaptive modulation process. Moreover, it is not clear whether the system can run multiple adaptive processes in parallel or not. Furthermore, the author didn't present how the partial bit-streams are transferred using ICAP or PCAP.

The authors in [DPML07] and [DMN<sup>+</sup>08] benefit from the PR technique to propose a reconfigurable radio system. In [DPML07], PR is applied dynamically to reconfigure a convolution encoder, a constellation mapper and an FIR filter. In [DMN<sup>+</sup>08], PR is applied in a Network on Chips (NoC) architecture for SDR applications. The authors test the proposed system with a demonstration based on a video transmission between two Virtex 4 FPGAs. The code rate and constellation mapper are dynamically reconfigured and the results are noticed by the quality of the video at the receiver. The aim of the research is to benefit from the PR technique to have the ability to use more resources of the FPGA by reconfiguring blocks to allow implementing different communication chains within the same area. The PR management is controlled by the MicroBlaze soft processor implemented on FPGA with an ICAP controller.

## 1.7 Summary

In this chapter, we have introduced the main concepts and the related works in the context of applying the partial reconfiguration technique on an ARM-FPGA platform for reconfigurable wireless systems. We have briefly described general wireless communication notions and introduced most popular wireless standards. Then, we have introduced the needs to reconfigure wireless systems and the concept of vertical handover. After that, we have described the PR technique on the ARM-FPGA platform.

Many of the related works have used the PR technique to create a reconfigurable radio system. Most of these works used the MicroBlaze soft processor with the ICAP interface. Also, it was not clear how the partial reconfiguration process is managed and controlled. In the context of the switching between multiple wireless systems using the PR technique we were not able to find any work that explains the management and the execution of the handover on the ARM-FPGA device.

In the next chapter we present an original ARM-FPGA based platform for reconfigurable wireless communication systems that benefits from the partial reconfiguration technique. The baseband functions of the wireless system are implemented within the FPGA fabric, whereas the reconfiguration of the different functional blocks is managed by processes running on the ARM processor. The role of these processes is to take decisions and manage the reconfiguration process of the concerned modules. A custom micro-kernel that is dedicated to partial reconfiguration is used to schedule and share resources among processes.

The main contributions of our work are :

- Proposing an ARM-FPGA platform for reconfigurable wireless systems using PR and adaptive technique.
- Proposing an ARM-FPGA platform for unified PHY layers for multiple wireless standards using PR technique.
- Proposing a Vertical handover algorithm based on neural networks and scoring system in the context of heterogeneous wireless communication networks.
- Quantitative Evaluation of the system performance based on a case study.
- Real time power measurements for PR architectures and operation using two SoC based boards.
- Demonstration was applied to illustrate the proposed concept based on specific case study.

ZedBoard, ZC706, and ZC702, which features an Zynq SoC at it's heart, were

the main platform used to retrieve the performance evaluation and apply the demonstrations.



## CHAPTER 2

# ARM-FPGA BASED PLATFORM FOR AUTOMATED ADAPTIVE WIRELESS COMMUNICATION SYSTEMS USING PARTIAL RECONFIGURATION TECHNIQUE

### 2.1 Introduction

Since the last decade, the concept of reconfigurable radio has been a hot topic. The adaptive techniques were some of the first proposed techniques that allowed applying modifications to both PHY or MAC layers of the wireless system. These modifications are performed according to certain conditions and the state of the communication channel within the overall network. The reconfiguration of wireless communication systems is an efficient solution for implementing adaptive techniques.

In this chapter, we present a Software/Hardware platform for automated adaptive wireless communication systems. The proposed system benefits from the partial reconfiguration technique that is available in the FPGA device. The baseband functions of the wireless system are implemented within the FPGA fabric, whereas the reconfiguration of the different functional blocks are managed by processes running on the ARM processor. The processes are based on theoretical adaptive algorithms. Their roles are to take decisions and manage the reconfiguration process

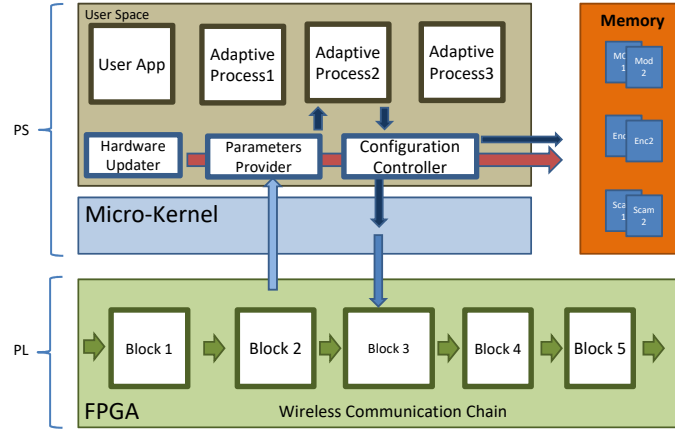


FIGURE 2.1 – General system design blocks

of the concerned modules.

## 2.2 General System Design

The proposed SoC system can be divided into two parts : the first part is running on the ARM processor, whereas the second part is executed in the FPGA logic. The general system design is presented in FIGURE 2.1.

### 2.2.1 Overview for ARM-FPGA platform

The ARM-FPGA platform is based on the Zynq-7000 device. This SoC integrates both PS and PL parts within the same chip. In this case, the PS uses the Device Configuration Interface (DevCfg) integrated on the ZYNQ device. The PCAP interface is driven by a 100 MHz clock and deals with a word width of 32 bits. The theoretical throughput is 3.2 Gbit/s, but due to the speed limit of transferring bit-streams from RAM, the interface speed is limited to 1.2 Gbit/s. In our case, we use the PCAP interface for its better compatibility with software applications and because it does not require additional logic resources in the FPGA.

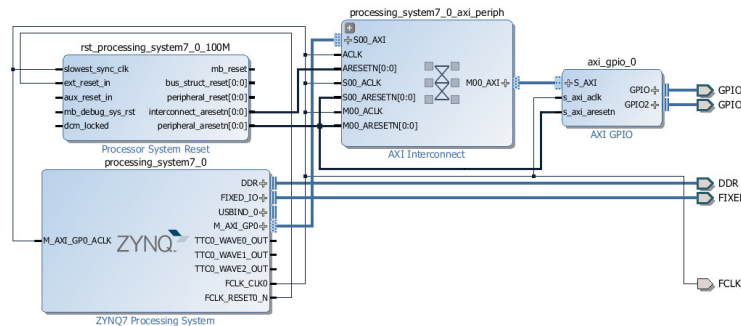


FIGURE 2.2 – Zynq block as represented in design flow using Vivado tools

## 2.2.2 Xilinx Vivado Tools

Vivado Design Suite is a software suite produced by Xilinx for synthesis and analysis of HDL designs, superseding Xilinx ISE with additional features for SoC development and high-level synthesis. Vivado represents a ground-up rewrite and re-thinking of the entire design flow (compared to ISE). Unlike ISE which relied on ModelSim for simulation, the Vivado System Edition includes an built-in logic simulator. Vivado also introduces high-level synthesis, with a toolchain that converts C code into programmable logic. Vivado enables developers to synthesize (compile) their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer. Vivado is a design environment for FPGA products from Xilinx, and is tightly-coupled to the architecture of such chips, and cannot be used with FPGA products from other vendors. FIGURE 2.2 illustrates the block design for Zynq with AXI interface as shown in Vivado.

Partial reconfiguration design flow on Vivado is more based on TCL commands than GUI where PR can be applied through GUI only in ISE.

## 2.2.3 Processing System Modules- PS part

In order to manage the partial reconfiguration on the ARM, the system is divided into 3 sub processes : The Configuration Controller, the Parameters Provider,



and the Hardware Updater. All these modules run as tasks in the user-space of the micro-kernel Ker-One. The role of the Configuration Controller applies PR on the PL and controls the data flow during the PR operation. The Parameters Provider provides the adaptive processes with needed parameters. The Hardware Updater adds or updates new versions of the RMs.

### **2.2.3.1 Micro-Kernel**

Ker-ONE is a lightweight micro-kernel that provides para-virtualization on ARM-based embedded systems. In this work, it is seen as a simple kernel that may implement several isolated tasks at user level. Since Ker-ONE is very simple, it only provides fundamental functions such as round-robin scheduling, inter-process communication (IPC) and memory management. It ends up with a small Trust Computing Base (TCB) as described in [XPN15]. In the proposed system, multiple adaptive processes run concurrently on the top of the kernel. Each task may access the DevCfg in order to reconfigure specific hardware blocks in the PL region of the FPGA. All the mechanisms that deal with the reconfiguration management are implemented in different isolated tasks : the configuration controller, parameters provider, and hardware updater.

### **2.2.3.2 Configuration Controller**

The configuration controller has two main roles. The first role consists in transferring partial bit-streams to the FPGA. For additional security reasons, the partial bit-streams memory locations are only accessible from the configuration controller. This controller eliminates any conflict that may occur when several user processes decide to access the DevCfg at the same time. In this case, it transfers the bit-streams consecutively. The adaptive processes does not manage the bit transfer operations but they only send requests and provide the memory addresses to the configuration controller that performs the operation.

The configuration controller module can also configure the concerned hardware modules by updating their input parameters in real-time. This operation is called parametric reconfiguration. In this case, the system reconfigures the hardware either by updating the input parameters of the concerned module or by sending a new version (bit-stream) of the RM. As an example, we can consider the IFFT module which can be reconfigured in real-time either by modifying the IFFT size parameter or by applying PR. This enables to combine both methods of reconfiguration (partial and parametric reconfiguration) to create a reconfigurable system.

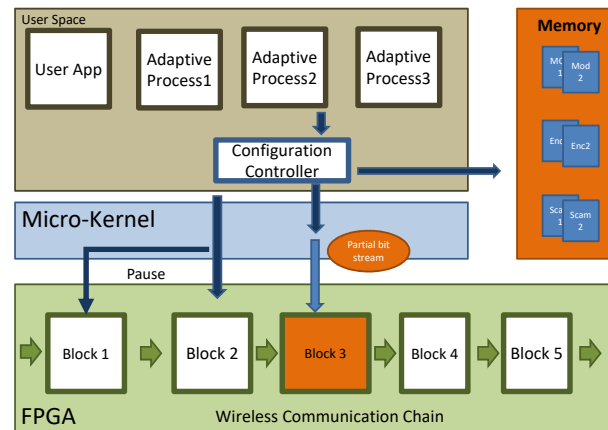


FIGURE 2.3 – Dataflow controlled by the Configuration Controller during Partial Reconfiguration

Choosing the configuration method depends on the architecture of the hardware module. Another important task of the configuration controller is to control the data flow during partial reconfiguration. When a FPGA chain is formed by a set of configurable modules, it is mandatory to make sure that the reconfiguration does not cause any loss of data in the RMs during the PR process. An example of the data flow management is described in FIGURE 2.3 where the second process (Process3) is requesting to modify the third module (Block3). In this case, the configuration controller suspends all the precedent modules (Block1 and Block2) that produce inputs to the reconfigured module (Block3). The modules are suspended by controlling the enable input signal in the module. During partial reconfiguration incoming data will be stored in FIFOs located between the blocks. After the module is reconfigured the precedent modules are enabled again and the data stored in the FIFOs are dequeued and processed. Note that the reconfiguration time of the concerned module is provided to the controller. This helps the controller to determine the time during which the modules should be suspended. The data flow of the hardware blocks is controlled through the AXI interfaces connecting the PS and PL.

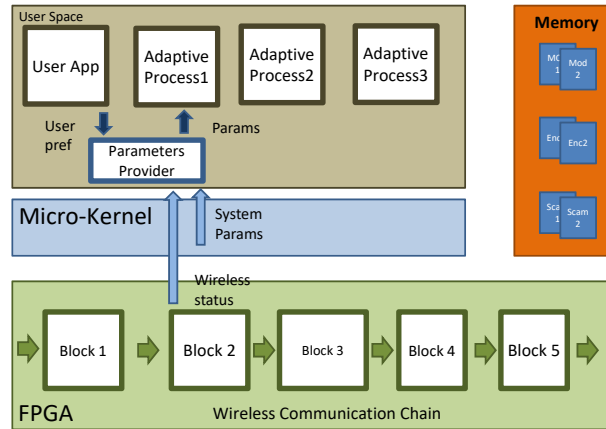


FIGURE 2.4 – Parameter Provider collects parameters from different layers

### 2.2.3.3 Parameters Provider

The parameters provider is another important task running on the micro-kernel. It is considered as an interface for all the parameters of the system. This module provides all the required parameters to the user processes in order to take a decision regarding the configuration of a specific module. The parameters are collected from different layers in the system. For example, the parameters provider collects parameters from physical layer such as SNR and BER, from the system parameters such as location, power consumption and battery level where these parameters can be also used to adapt the wireless chain.

### 2.2.3.4 Hardware Updater

The third module running on the processor is the Hardware Updater. In this module, we benefit from one of the advantages of the PR that makes it possible to implement new versions of modules or even update other available modules during runtime. In a partial reconfiguration system design, RMs are defined as black boxes with predetermined input and output ports. This means that all the versions of a RM must have the same input and output ports. Many partial bit-streams can be generated but only one is implemented on the FPGA at a time. The other partial bit-streams are transferred to the RM on demand. In such designs, we have the

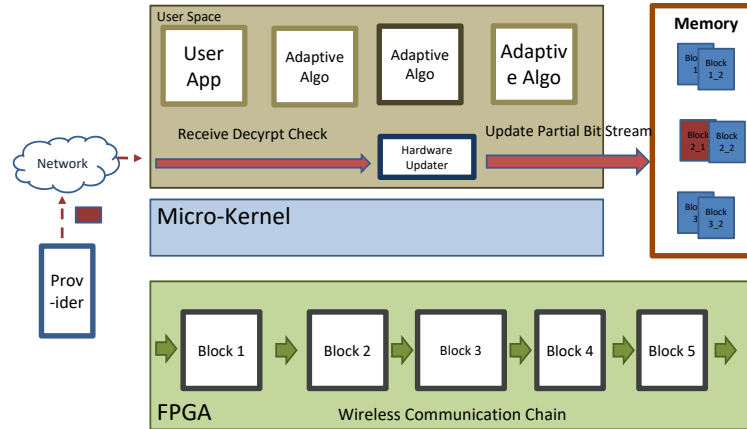


FIGURE 2.5 – Hardware Updater updates versions of RMs

possibility to reconfigure any RM without reconfiguring all the hardware chain. The task of the hardware updater is to check for any new update or new version of a RM. If any update is available, a module must retrieve its new partial bit-stream in a secure way, which means that the update server should authenticate the user requesting the update. Also the partial bit-stream should be encrypted while transmitted on the channel. The idea is similar to updating or sending patches to a software process, but in this case it are applied to hardware. The update or upgrade for the RM will be applied over the network, so the transmitted files should be encrypted and signed from the update server. This will protect the update process and ensure data integrity. At the receiver side, the system decrypt the received file then check if it is signed by the server. After that the received files is transferred to memory to upgrade versions of the partial bit-streams or to add a new version for a specific RM. These new versions can then be used for further PR.

## 2.3 Auto Adaptive OFDM transmitter

To study the impact of applying partial reconfiguration in the context of reconfigurable radios, we have implemented an auto-adaptive OFDM transmitter in the proposed system.

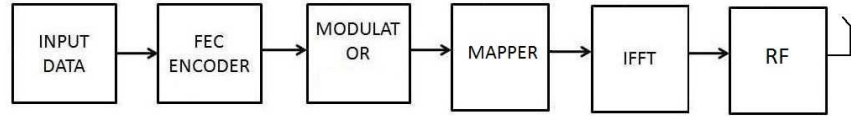


FIGURE 2.6 – OFDM transmitter base modules

### 2.3.1 Reconfigurable Hardware

The generic OFDM transmitter is composed of an encoder, a modulator, a mapper, and an IFFT (see FIGURE 2.6). Based on the Xilinx Vivado tool, multiple partial bit-streams have been generated for these modules. The different versions of the RMs and the parameters used in the transmitter are given in TABLE 2.1.

TABLE 2.1 – OFDM transmitter parameters

Parameters	Scenario
Modulation	4-QAM,8-QAM,16-QAM,32-QAM,64-QAM
(I)FFT Size	16,32,64,128,256,512,1024
Cyclic Prefix Length	2,4,8,16,32,64,128
Encoder	Convolutional 1/2,1/3,1/4,1/5, 3GPP LTE Turbo Encoder 1/3
Frame	1 OFDM pilot + 10 OFDM data symbols
FPGA	Zynq-7000 xc7z020
Clock	200 MHZ

The OFDM transmitter has been implemented and tested on the ZedBoard. The data path operates on the FPGA as follows : A random bit source generator provides random data to the encoder. Then, data are encoded and provided to the modulator in which frequency domain symbols are generated. The symbols are mapped into sub-carriers in the IFFT module through the mapper. Finally, ten OFDM symbols are generated by the IFFT module followed by cyclic prefix and then concatenated with pilot bits to build a frame. The frame is then ready to be transferred to the Radio Frequency (RF) module. The reconfigurable regions on the FPGA (Pblocks) are selected using the floor-planning tool. Selecting a Pblock with an amount of resources larger than necessary leads to wasting resources on the

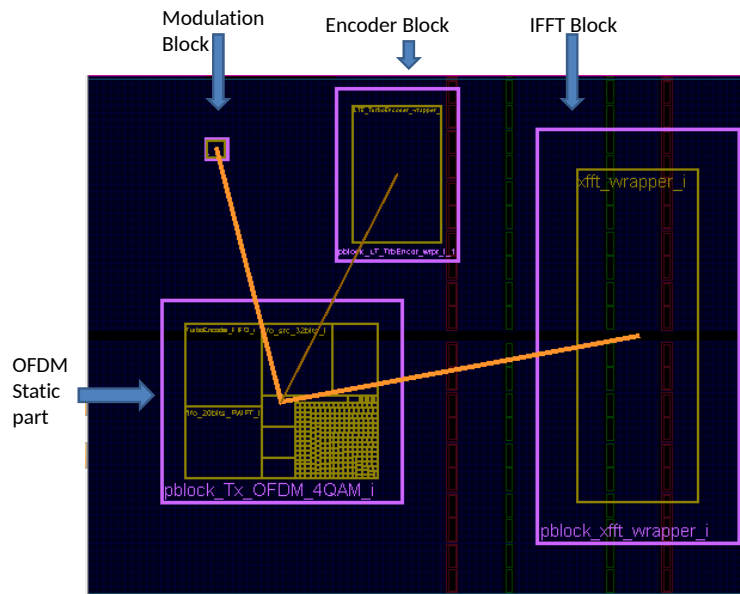


FIGURE 2.7 – Floorplan view of the reconfigurable and static modules of the OFDM transmitter

FPGA. Therefore, the selected Pblocks are tuned to the maximum resource utilization that is required by the different versions of the RMs. FIGURE 2.7 shows the floorplan design of the reconfigurable and static modules of the OFDM transmitter. As required in a PR design, the different versions of a given RMs should have the same interface (input and output ports). For example, if a 4-QAM modulator has a 2-bit input whereas a 256-QAM modulator has an 8-bit input, then the reconfigurable modulator will match the largest module with 8-bit input. Therefore, 6 bits will be unused in the data input port of the 4-QAM. A similar concept is applied to all the RMs in the chain. Note that these additional bits have no impact since they are likely to be not considered in the optimization process of the design flow. FIGURE 2.8 presents the adaptive OFDM transmitter implemented in the proposed platform. The modulator, encoder, and IFFT blocks are RMs whereas the other blocks remain static. The partial bit-streams corresponding to multiple versions of each RM are located in memory, and transferred to FPGA on demand by the configuration controller. Note that in some cases, such as for the (I)FFT architectures, the concerned module can be reconfigured by parametric reconfiguration without initiating the DPR process and transferring a bit-stream from memory. This can be done by simply modifying the input size parameter of the

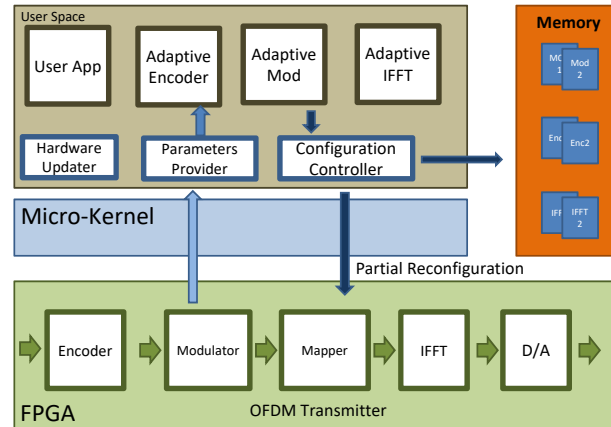


FIGURE 2.8 – Adaptive OFDM transmitter implemented in the proposed design module. This situation is also taken into account in the configuration controller.

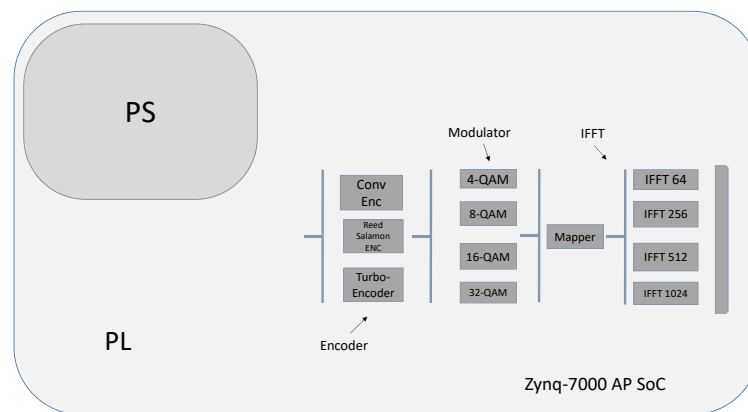


FIGURE 2.9 – Parallel implementation for reconfigurable OFDM with multiple blocks

In FIGURE 2.9 and FIGURE 2.10, the difference between the parallel and PR implementation designs in the OFDM scenario is illustrated. In the PR design, the different versions of the RMs stored in memory and loaded on demand. On the other hand in the parallel design all the blocks are implemented in parallel

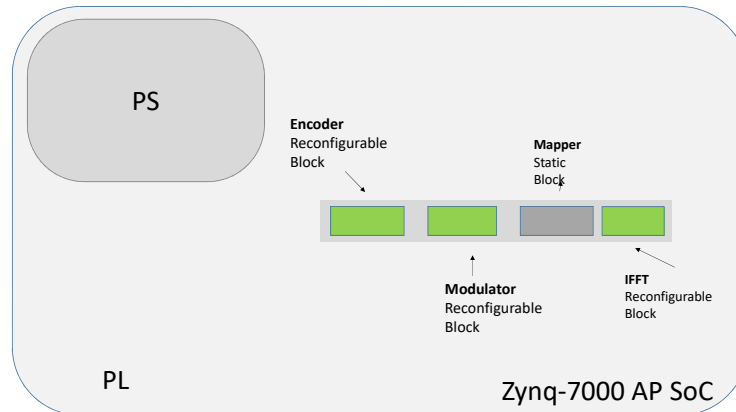


FIGURE 2.10 – Implementation for reconfigurable OFDM with multiple blocks using PR technique

and data flow is modified by switching between the different static modules during runtime.

### 2.3.2 Adaptive processes

The Xilinx SDK tool has been used to implement three adaptive processes running on the ARM processor. The processes are implemented in C code and run in parallel on top of the micro-kernel Ker-One. Adaptive processes has the task to request reconfiguration for a specific RMs in the wireless chain implemented on FPGA. In this section, three adaptive processes are described in the selected scenario.

The first process  $(adaptive\_mod)$ , is based on the adaptive modulation technique and aims to monitor the SNR value to select the best modulation scheme. The role of this process is to reconfigure the modulation block during run-time. The implemented  $adaptive\_mod$  process can be explained in FIGURE 2.11. The process is first initialized (for example with a value corresponding to a 4-QAM modulation) then it waits until another SNR value is received. When the process receives a new SNR value, it is analyzed to check if it belongs to the current range. If this is the case, the process returns to the waiting state. Otherwise, a new version of the modulation block is selected according to the new range. The partial bit-stream of the selected version is then transferred to the modulator. Finally, the current operation of the SNR range is updated and the process waits for another SNR value. In Listing 2.1, the pseudo code running on the ARM for the adaptive



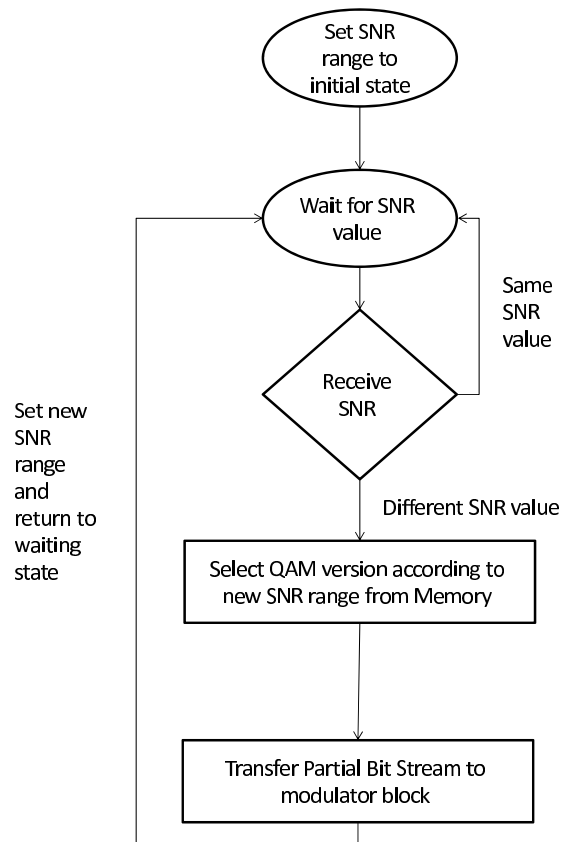


FIGURE 2.11 – Flow chart for the adaptive modulation process

modulation process is also shown.

The second process, *adaptive\_encoder*, is based on the adaptive channel coding technique. This process reads the BER and SNR values to obtain the wireless channel status in order to select the best coding rate or the most efficient encoder type. This second process reconfigures the encoder module of the OFDM chain. The purpose of the third process, *adaptive\_IFFT*, is to modify the IFFT size according to the power consumption and remaining energy in the battery of the system. This process monitors the power consumption in order to select the most efficient IFFT size.

In Listing 2.2, the C code implementation of the function of the Configuration Controller is presented, where the data flow is controlled before and after the partial reconfiguration process. It is shown how the Configuration Controller take into consideration the order of the blocks in the communication chain to know

which block should be suspended during the PR operation.

The reconfiguration settings at the transmitter side (modulator, encoder, fft size) is considered to be transmitted the receive side by control signals. The reconfigurable modules are then reconfigured according to the received settings.

The three adaptive processes are executed as user processes. In this work, we have considered a realistic scenario in which all the three processes are based on theoretical adaptive techniques and reconfigurable wireless system methodologies as proposed in [FS], [DF03], and [ZEA05].

Listing 2.1 – Pseudo code for adaptive modulation process

```

1
2
3 load PARTIAL_RECONFIG_Mod_QAM8_Bit_STREAM
4 load PARTIAL_RECONFIG_Mod_QAM16_Bit_STREAM
5 load PARTIAL_RECONFIG_Mod_QAM32_Bit_STREAM
6 load PARTIAL_RECONFIG_Mod_QAM64_Bit_STREAM
7
8
9 while true:
10     read SNR;
11
12     if SNR less than 15:
13
14         if status NOT EQUAL to QAM8:
15
16             Pause Encoder_Block
17             Send PARTIAL_RECONFIG_Mod_QAM8_Bit_STREAM
18             Un_Pause Encoder_Block
19
20     else if SNR less than 18:
21
22         if status NOT EQUAL to QAM16:
23
24             Pause Encoder_Block
25             Send PARTIAL_RECONFIG_Mod_QAM16_Bit_STREAM
26             Un_Pause Encoder_Block
27
28     else if SNR less than 21:
29
30         if status NOT EQUAL to QAM16:
31
32             Pause Encoder_Block
33             Send PARTIAL_RECONFIG_Mod_QAM16_Bit_STREAM
34             Un_Pause Encoder_Block
35
36     else if SNR less than 24:
37
38         if status NOT EQUAL to QAM32:
39
40             Pause Encoder_Block
41             Send PARTIAL_RECONFIG_Mod_QAM32_Bit_STREAM
42             Un_Pause Encoder_Block

```

```

43
44     else
45
46         if status NOT EQUAL to QAM64:
47
48             Pause Encoder_Block
49             Send PARTIAL_RECONFIG_Mod_QAM64_Bit_STREAM
50             Un_Pause Encoder_Block
51
52     end while

```

Listing 2.2 – Configuration Controller C code

```

1 // Configure Function with dataflow control
2
3 int configure(int order,int version)
4 {
5     while (XDcfg_IsDmaBusy(XDcfg_0) == XST_SUCCESS)
6     {
7         xil_printf("Driver Busy : Wait to finish!\n\r\n\r");
8     }
9     int i;
10    for (i=1;i<order;i++) // pause all previous blocks
11    till transfer bit stream finish
12    {
13        #ifdef gpio
14        XGpioPs_WritePin(&Gpio, i, 1);
15        #endif
16    }
17
18    int Status = XDcfg_TransferBitfile(XDcfg_0,
19    array_of_bits[order].address[version],
20    (array_of_bits[order].length >> 2));
21    if (Status != XST_SUCCESS) {
22        xil_printf("ERROR : FPGA configuration failed!\n\r\n\r");
23        //exit(EXIT_FAILURE);
24    }
25
26    for (i=1;i<order;i++) // unpause all previous blocks
27    after transfer bit streams finish
28    {
29        #ifdef gpio
30        XGpioPs_WritePin(&Gpio, i, 0);
31        #endif
32    }
33 }

```

## 2.4 Experimental Results

This section presents the results obtained from the adaptive OFDM transmitter scenario. The results are described in terms of the time needed to reconfigure the

RMs, partial bit-streams size, power consumption and reserved resources by each version of the modules. Finally, a Gantt chart is presented to illustrate the behavior of the system according to the input parameters' variations. From the obtained results, we aim to show how much hardware resources are used and evaluate the corresponding power consumption as well as the reconfiguration time.

In the adaptive OFDM scenario, we consider three RMs : the modulator, encoder and IFFT. The results presented in TABLE 2.2 include the partial bit-stream size and the time required to reconfigure a module. As mentioned earlier, the size of the partial bit-stream is related to the size of the RM implemented in the FPGA. This size is selected to fit the largest version of the concerned RM among all the possible ones.

TABLE 2.2 – Partial bit-streams size and reconfiguration time

Module	Partial Bit-Stream Size	Reconfiguration Time
IFFT	188 KB	1.48 ms
Encoder	106 KB	0.834 ms
Modulation	30 KB	0.236 ms

As noticed in TABLE 2.2, the time needed to transfer a bit-stream to hardware using the PCAP interface depends on the size of the partial bit-stream file. In our work, the reconfiguration time is computed by counting all the clock cycles needed to complete the partial bit-stream transfer. Then, the number of clock cycles is multiplied by the clock period (1/Frequency of processor). The way to compute the partial reconfiguration time is shown in Listing 2.3.

Listing 2.3 – Computation of partial reconfiguration time needed

```

1 u32 timelow =0;
2 u32 timeaft =0;
3 u32 time ;
4 double counts = 0.002999995;
5 timelow = XTime_GetLow();
6 SD_TransferPartial("pr500.bin", PARTIAL_RECONFIG_1_ADDR, PARTIAL_BITFILE_LEN_1);
7 timeaft = XTime_GetLow();
8 time=(timeaft-timelow) * counts;

```

The partial bit-stream size is related to the floorplan design, as shown in FIGURE 2.7, where the IFFT module occupies more resources on the FPGA compared to the modulation block.

Vivado power and utilization estimation tools have been used to estimate the power consumption and resources used by the different versions of the RMs. Vivado

power tools are only used to estimate static power for different versions of each module. These results are then used to compare the static power of the PR-based design with the parallel-based one. For both cases, the dynamic power has not been taken into account since it is closely related to the input stimuli provided during timing simulation. These stimuli are similar for both the reconfigurable and parallel architectures. The architecture of the IFFT used in this scenario is the pipeline streaming IFFT (from the Xilinx IP library). This IFFT module allows performing a continuous computation.

TABLE 2.3 – Size and power consumption of the considered blocks

Module	Power (mW)	FPGA Resources (LUT-REG-RAM-DSP)
4-QAM	less than 0.1	15 - 8 - 0 - 0
8-QAM	less than 0.1	16 - 10 - 0 - 0
16-QAM	less than 0.1	18 - 15 - 0 - 0
32-QAM	less than 0.1	20 - 19 - 0 - 0
64-QAM	less than 0.1	22 - 20 - 0 - 0
Convolution 1/2	less than 0.1	24 - 29 - 0 - 0
Convolution 1/3	less than 0.1	25 - 31 - 0 - 0
Convolution 1/4	less than 0.1	26 - 32 - 0 - 0
Convolution 1/5	less than 0.1	28 - 35 - 0 - 0
Turbo-Encoder 1/3	2	311 - 555 - 0 - 0
16-point IFFT	9	191 - 1299 - 0.5 - 3
1024-point IFFT	29	2245 - 3554 - 3 - 12

TABLE 2.3 shows the differences between the IFFT, encoder, and modulation blocks in terms of utilized resources and power consumption. These resources and the on-chip power consumption are almost identical for the different modulation versions. This observation is also true for the convolution encoders. The Turbo encoder clearly consumes much power and occupies more resources on FPGA than the convolution encoders do. As for the different versions of the IFFT block, it is obvious that the power consumption and the utilized resources increase as the IFFT size increases.

Also, it is important to analyze the running processes through a Gantt chart as depicted in FIGURE 2.12. As shown in this figure, three adaptive processes run in parallel on the PS part, sense their own parameters and finally detect any change. If necessary, the partial reconfiguration is applied on the appropriate block in order to adapt the system to the new environment.

As an example, we consider the scenario illustrated in FIGURE 2.12. During initialization, a default configuration is implemented in the FPGA : an 8-QAM modulator, a turbo-encoder and a 1024-points IFFT.

At time  $t = t_1$ , the *adaptive\_mod* process detects a high SNR. According to the algorithm described for the *adaptive\_mod* process, a partial reconfiguration is launched by the configuration controller that leads to implementing a 64-QAM in order to increase the throughput. The partial reconfiguration is done and a new configuration is effective at time  $t = t_1 + 0.236$  ms. Meanwhile the encoder block is suspended for 0.236 ms.

At time  $t = t_2$ , the *adaptive\_coding* process detects a decrease in the BER. Therefore, a partial bit-stream of the convolution encoder is transferred to the encoder module. The transfer is done at  $t = t_2 + 0.834ms$ . Since the encoder is the first module in the chain, the other modules keep on running during the partial reconfiguration operation.

At  $t = t_3$ , the *adaptive\_FFT* process that senses the power regulators on the device detects a high power consumption with a low battery voltage level. Therefore, this process reconfigures the IFFT module and reduces its size. This leads to reducing the power consumption of the system. The PR operation takes 1.48 ms and all the precedent modules are suspended during the partial bit-stream transfer operation in order to avoid any data loss. It may be noticed that the time needed to reconfigure the IFFT module is longer than those of the coding and modulation modules. This is due to the difference in the size of the partial bit-streams. The suspension of the precedent modules is clearly shown in FIGURE 2.12. The suspension time depends on the size of the transferred partial bit-streams. As noticed, the suspension overhead only affects the precedent modules of the chain.

As a result of applying partial reconfiguration, the FPGA power consumption and reserved resources are reduced. As an alternative to this technique, all versions of the different modules may be implemented in the FPGA, with additional multiplexers to select the desired module. This typically requires a lot of resources on FPGA. As a consequence, the static and dynamic power consumption of the chip would be relatively high compared to PR-based systems.

TABLE 2.4 shows the amount of resources that is used when applying the PR technique compared to the case where PR is not adopted. It may be seen that there is approximately a factor of 4 between the two cases. Moreover, in the PR case, the resource gain makes it possible to use the remaining parts of the FPGA to implement additional processing units. The percentage of the LUTs, REGs and DSPs used on the SoC is 19.3%, 15.9%, and 25.9%, respectively when all blocks

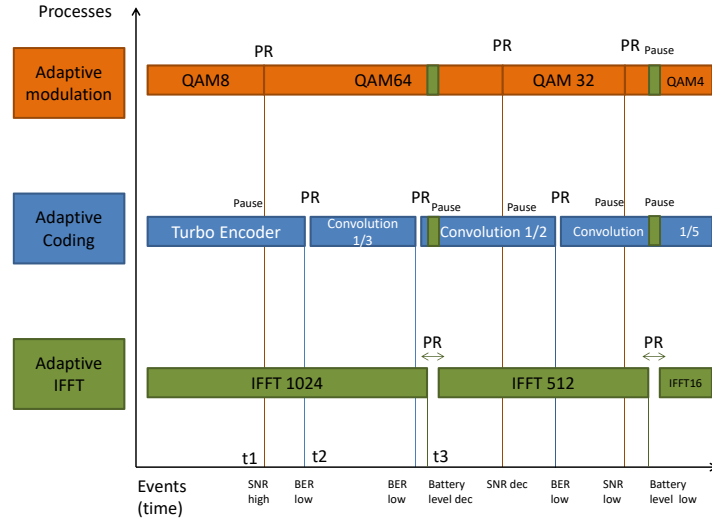


FIGURE 2.12 – Adaptive processes reconfiguring OFDM transmitter during run-time according to values of parameters

TABLE 2.4 – Comparison of hardware resource usage

Resources	LUTs	REGs	DSPs
with partial reconfiguration	2578	4129	12
without partial reconfiguration	10320	16938	57
Available on ZYNQ	53200	106460	220

are implemented in parallel. When using PR technique, the percentage of the corresponding used resources is 4.8%, 3.8%, and 5.45%, respectively.

## 2.5 Summary

In this chapter, an ARM-FPGA platform has been proposed for an adaptive wireless communication system. The system implemented on the FPGA is partially reconfigured. This HW/SW platform is based on a custom micro-kernel that is already available in our laboratory. Its role consists in managing and controlling partial reconfiguration processes. Multiple software processes based on theoretical adaptive algorithms run in parallel and perform partial reconfiguration operations. These processes get knowledge of the environment condition by requesting

parameters from the Parameters Provider. In this work, we made extensive use of the PCAP interface and optimized the size of the partial bit-streams in order to minimize the time overhead caused by partial reconfiguration. Also, to avoid data loss, the Configuration Controller controls the data flow on the FPGA when initiating the PR process. An adaptive OFDM transmitter was implemented to test the proposed system. The obtained results have shown that implementing the PR technique with the micro-Kernel has reduced both the global on-chip power consumption and reduced the reserved resources on the FPGA. This study gives an important information about the impact of adopting the PR technique in the context of reconfigurable wireless communication systems.

After proposing the SoC system for auto adapted reconfigurable radios, it was interesting to propose a similar platform for a reconfigurable unified PHY layer for multiple wireless standards. In the next chapter, we will discuss, using PR technique, an ARM-FPGA platform for switching between multiple wireless standards. The proposed system is described, a case study is considered to evaluate the performance of the proposed system and to obtain numerical results.





## CHAPTER 3

# RECONFIGURABLE UNIFIED PHY FOR MULTI WIRELESS SYSTEMS USING PARTIAL RECONFIGURATION

### 3.1 Introduction

In the previous chapter we have described the proposed ARM-FPGA system for auto adaptive reconfigurable radios using PR technique. It is also interesting to use the ARM-FPGA system to switch between multiple wireless systems. In this chapter we will describe how the ARM-FPGA system can be adopted to handle switching between multiple wireless standards. On the PL part, PR is used to reconfigure a unified wireless chain to switch between multiple wireless standard. For this purpose, two different proposed reconfigurable architectures are compared. Finally, a unified reconfigurable wireless chain for WiFi-WiMax is tested as a case study to extract and evaluate the experimental results.

### 3.2 ARM-FPGA Platform for Switching between Multi Wireless Systems

This section describes the design of a reconfigurable PHY layer architecture of multi wireless systems. As depicted in FIGURE 3.1, the system is divided into two parts : software and hardware. In the software part, processes run on the ARM processor to control and manage the configuration of the wireless chain

implemented on the hardware. In the hardware part, the system is implemented and designed to be partially reconfigured during runtime.

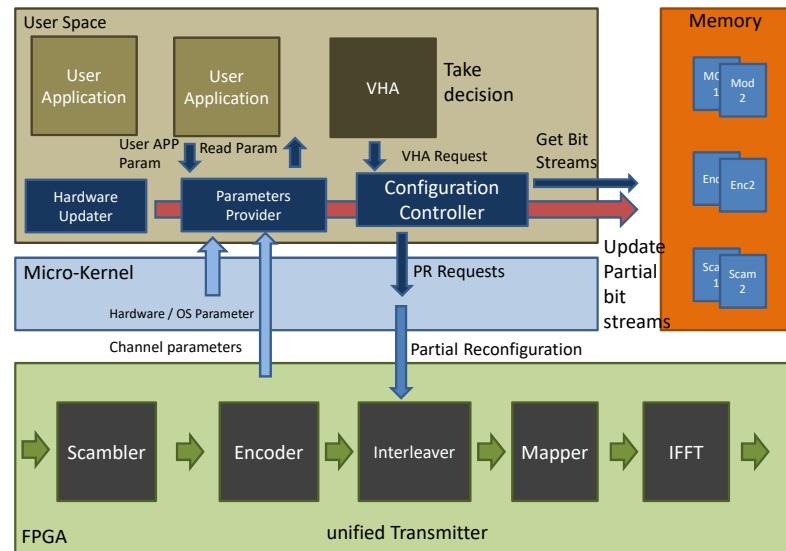


FIGURE 3.1 – General system design blocks in the scenario of unified chain for multiple wireless systems

### 3.2.1 Software Part

The software part of the system runs in the user space of the Ker-ONE microkernel dedicated for the partial reconfiguration management. Three modules, similar to PS modules proposed in CHAPTER 2, form the main architecture of the software part in addition to the VHA. The role of the VHA is to read parameters from the available communication systems, learn, and decide whether to switch to another system or to keep on operating with the current one.

The Parameters Provider has the task to provide the VHA with all the available parameters that are necessary to make a correct decision. During run-time, the Parameters Provider module collects the values of the parameters from different layers. For example, channel status parameters like the SNR, RSS, and BER

are collected from the hardware part on the FPGA. User preference parameters (throughput, real time demands, low cost communication, low power communication, high security connection, ...) are collected from processes running in the user space. Furthermore, the Parameters Provider reads parameters from the hardware and software parts such as the power consumption, battery level, speed of the node as well as the location information. The VHA process manages the requests to switch from a standard into another. The configuration controller manages and controls the partial reconfiguration of the RMs in the FPGA. The processor uses the DevCfg driver to reconfigure the FPGA through the PCAP interface.

In CHAPTER 2, it has been explained how the configuration controller works in details in the case of reconfigurable radio scenario. In the current scenario, when switching the communication between multiple standards, the Configuration Controller block has the task to configure the unified chain when the system decides to switch from a standard into another. The Configuration Controller manages the data flow during the reconfiguration process.

In this scenario, the Hardware Updater has the ability to update a block of the RMs by transferring partial-bit-streams to the memory. Also, it has the possibility to add a new wireless standard if it is compatible with the reconfigurable unified chain on the hardware.

The data flow of the software part can be summarized as follows : The Parameters Provider provides the VHA with the needed parameters from different layers in the system. The VHA analyzes the values of the parameters and makes a decision regarding a possible and useful handover. Accordingly, the VHA sends a request to the configuration controller to switch to another standard. Then, the Configuration Controller manages the transfer of the partial bit-streams to the RMs on the FPGA through the PCAP interface.

### 3.2.2 Hardware Part

The hardware part of the system consists of physical layers of communication systems implemented on the FPGA. Using partial reconfiguration, we aim to implement a single unified reconfigurable chain for different wireless standards. As an example, FIGURE 3.2 depicts the switching between two wireless standards on the FPGA. In state1, the Sys1 blocks are running while the partial bit-streams of the Sys2 blocks are stored in memory. When VHA decides to switch to Sys2, partial bit-streams of Sys2 are transferred to replace the Sys1 blocks. The communication system is supposed to run in parallel with other hardware accelerators on the

FPGA. When partial reconfiguration is applied, the other accelerators continue running. The unified chain for multiple wireless standards can be implemented in two possible reconfigurable architectures. These architectures will be described in the next section. Also, a comparison between these two architectures is discussed in the next section.

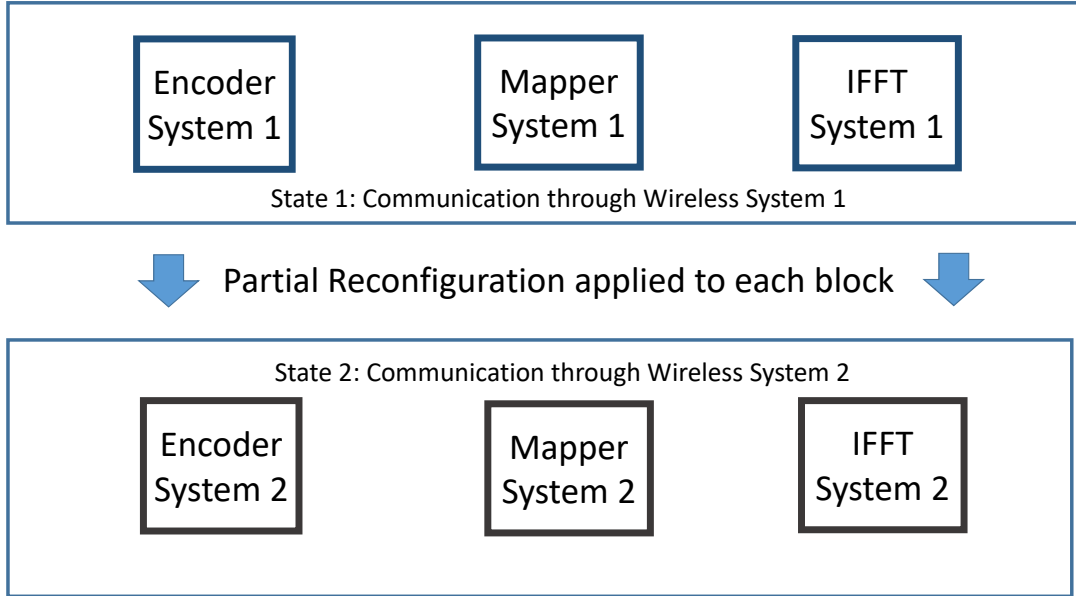


FIGURE 3.2 – State of part of the RMs during VHA

### 3.3 Proposed Reconfigurable Architectures

On the PL part, a reconfigurable unified physical layer for multiple wireless systems is implemented. In this section, two reconfigurable architectures using PR are proposed and compared in terms of reconfiguration time, design complexity, and functionality.

In the first architecture, the entire communication chain is considered as one RM. In this case, each wireless standard has only one partial bit-stream. FIGURE 3.3 provides an example related to the switching process. For  $n$  systems,  $n$  partial bit-streams are stored in memory. At runtime, only one partial bit-stream is transferred to the reconfigurable block. When the system decides to switch to another wireless system, the desired partial bit-stream is transferred to replace the

TABLE 3.1 – Comparison between both reconfigurable schemes

Specifications	One Reconfigurable Unified Chain Architecture (ORBA)	Multiple Reconfigurable Blocks Architecture (MRBA)
<b>Reconfiguration time</b>	The time needed to transfer the partial bit-stream of the unified chain.	The time needed is the sum of the reconfiguration times of all the reconfigurable blocks.
<b>Partial bit-stream</b>	<ul style="list-style-type: none"> <li>- One partial bit-stream is transferred during reconfiguration.</li> <li>- Large bit-stream.</li> </ul>	<ul style="list-style-type: none"> <li>- <math>m</math> partial bit-streams are transferred during reconfiguration.</li> <li>- Small bit-streams.</li> </ul>
<b>Resources Utilization</b>	Almost same resources reserved for all architectures.	Almost same resources reserved for all architectures.
<b>Design Complexity</b>	<ul style="list-style-type: none"> <li>- Easier in partial reconfiguration design.</li> <li>- No ability to have shared blocks as a static part</li> </ul>	<ul style="list-style-type: none"> <li>- Harder is Partial Reconfiguration Design.</li> <li>- Possibility to have shared blocks as a static part in the communication chain.</li> </ul>
<b>Hardware Update</b>	<ul style="list-style-type: none"> <li>- All chain is updated when one block is needed to be modified.</li> <li>- Size of updated bit-stream is relatively larger.</li> </ul>	<ul style="list-style-type: none"> <li>- Possibility to update one reconfigurable block.</li> <li>- Size of updated bit-stream is relatively smaller.</li> </ul>
<b>Runtime Reconfiguration Complexity</b>	<ul style="list-style-type: none"> <li>- Simple reconfiguration process.</li> <li>- The data flow from upper layer is buffered until partial reconfiguration is done.</li> </ul>	<ul style="list-style-type: none"> <li>- More complex reconfiguration process.</li> <li>- Blocks are reconfigured sequentially and some of them are suspended until all blocks are reconfigured.</li> </ul>

existing one. This reconfigurable architecture is denoted as One Reconfigurable Block Architecture (ORBA).

The second architecture, called Multiple Reconfigurable Blocks Architecture (MRBA), divides the communication system into multiple reconfigurable blocks. In this case, for each reconfigurable block in the chain,  $n$  partial bit-streams are stored in external memory, where  $n$  is the number of considered wireless standards. And, if each chain is divided into  $m$  reconfigurable modules, then  $(m * n)$  partial bit-streams are needed. FIGURE 3.2 shows an example on the multiple reconfigurable blocks architecture. Both reconfigurable schemes are compared to

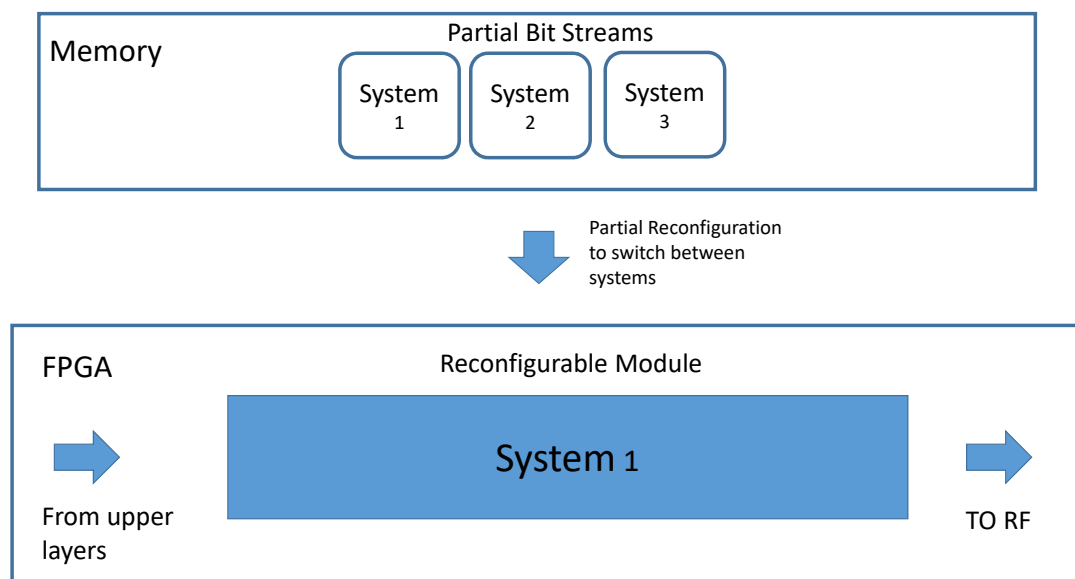


FIGURE 3.3 – Partial reconfiguration of the the unified chain using ORBA

define the characteristics of each architecture. The advantages and disadvantages of each architecture allows to determine which architecture is better to adapt in a given application. In terms of design complexity, ORBA is considered as easier to design than MRBA. In MRBA, studying the similarities and differences of the blocks forming the multiple wireless systems is needed. This study is essential to determine which blocks are reconfigurable and which ones are static. The static blocks are considered as common and thus they can be used in all the systems without the need of reconfiguration. It is not the case in ORBA in which all the blocks of a wireless system are grouped to form one reconfigurable block with an interface to the upper layer. In this case, it is easier to implement the system using

partial reconfiguration. In ORBA, the possibility to have static modules in the chain (if they exist) is discarded. Both architectures have quite similar resources utilization in the FPGA. In MRBA, the reconfiguration time is the sum of the reconfiguration times of the concerned reconfigurable blocks. The time needed to transfer a partial bit-stream to the FPGA is directly related to its size. The size of the reconfigurable blocks varies according to their complexity. In ORBA, only one partial bit-stream is transferred in order to switch between systems. Since the unified chain contains all the blocks, the size of the partial bit-stream is relatively large compared to the case of MRBA where the partial bit-streams represent relatively small RM. In most cases, there are many shared blocks between wireless standards, and some blocks can be configured by only changing input parameter values (ex. size of IFFT). This will reduce the number of reconfigurable blocks. In this case, the reconfiguration time in ORBA is greater than that in MRBA. To update one reconfigurable block in the chain when using ORBA, the whole wireless chain must be updated. On the other hand, it is possible to update only one reconfigurable block when using MRBA. Thus, the updated PR bit-streams are relatively large when dealing with ORBA. During runtime, switching from one system to another is simple when using ORBA. The data flow from the upper layer is buffered until the partial reconfiguration is done. On the other hand, switching in MRBA is more complex since all the reconfigurable blocks are partially and sequentially reconfigured. In addition to buffering the incoming data from the upper layer, and during the reconfiguration of each block, the blocks preceding the ones being reconfigured are suspended until all the PR blocks are successfully reconfigured. TABLE 3.1 summarizes the comparison between the two schemes.

### 3.4 Case Study : WiFi-WiMax Unified Reconfigurable chain

In this section, the proposed ARM-FPGA system is tested on a WiFi-WiMax scenario. The physical layers for the two standards are compared and a unified reconfigurable chain is proposed based on the partial reconfiguration technique. In this section, we present the physical layer for each standard, then we explain the proposed unified chain.



### 3.4.1 IEEE 802.11 WiFi

The first IEEE specification for WLAN was released in 1997. Today, millions of IEEE 802.11 devices are in use around the world. The IEEE has created different releases of the 802.11 standard. The earlier legacy release was based on the Direct Sequence Spread Spectrum (DSSS) technique. The 802.11a,g,n,ac releases were based on the OFDM technique. In 2009, the Multiple Input Multiple Output (MIMO) technique was added in the 802.11n to provide higher data rate and better performance. In our work, we have adapted the physical layer proposed in 802.11a. The main blocks in the physical layer of the IEEE 802.11a release transmitter are shown in FIGURE 3.4 and the explanation of each block is written in the yellow boxes. WiFi 802.11a supports shared data rate up to 54 Mbit/s with a coverage range up to 100 meters.

### 3.4.2 IEEE 802.16 WiMax

The IEEE 802.16 WiMax standard was created to provide better range than the WLAN standards. For that, the physical layer, bandwidth and carrier frequency were different from those in WiFi. Even though, the physical chain of WiMax has similarities with that of WLAN standard since it is also based on the OFDM technique. WiMax supports shared data rate up to 70 Mbit/s and coverage range of up to 10 km. In FIGURE 3.4, the blocks of the WiMax PHY chain are explained in the green boxes.

### 3.4.3 Unified chain Design using PR

After providing a brief overview of the two famous wireless standards (WiFi and WiMax), we aim now to create an unified reconfigurable communication chain for both standards using the PR technique. In this section, we study the similarities and differences in the physical layer of the two standards. A unified reconfigurable architecture based on PR is then proposed and described in details. The corresponding design flow is also presented.

IEEE defines the PHY and MAC layers specifications for both WiFi and WiMax. The physical layer of these two standards is based on the OFDM modulation scheme. The main modules are : Scrambler/Descrambler, Forward Error Correction (FEC) Encoder/Decoder, Interleaver/De-Interleaver, Mapper/De-Mapper, Pilot Guard Insertion/Removal, IFFT/FFT, and Cyclic Prefix Insertion/Removal. The differences in the transmitter architectures are illustrated in FIGURE 3.4.

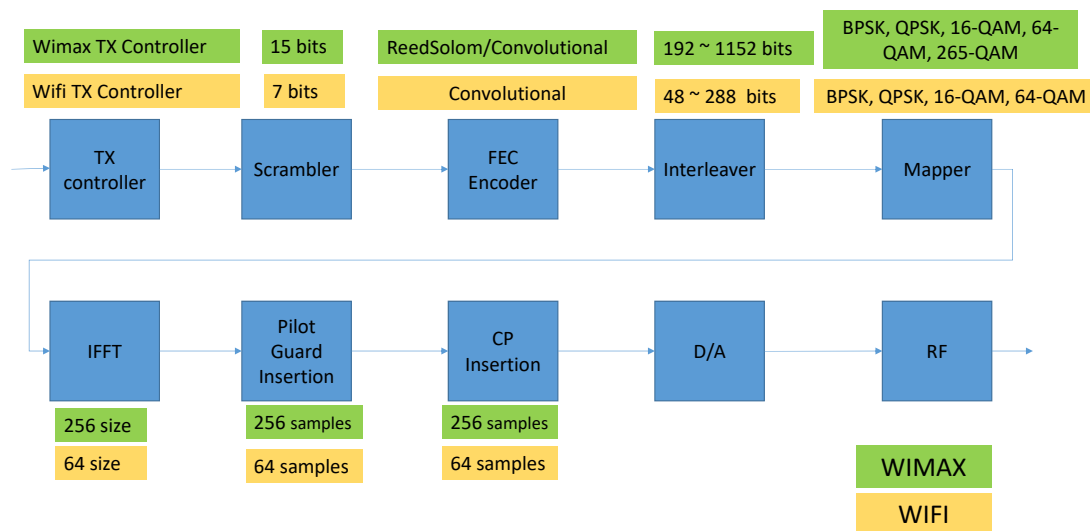


FIGURE 3.4 – State of the RMs during VHA

As shown in FIGURE 3.4, the PHY layer architectures of both standards are quite similar with few differences in the functionality of some modules. For example, WiMax uses a Reed-Solomon FEC with a convolution encoder whereas WiFi uses the convolution encoder only. The differences in the receiver are similar to those in the transmitter. Thus, it seems suitable to apply PR with the MRBA method.

Applying PR in the system design using MRBA requires to divide the system into two parts, static and dynamic. The static part consists of shared blocks used in both standards. During the switching process between standards, this part of the system remains unchanged. The dynamic part is composed of modules that can be reconfigured when necessary. Multiple versions of these RMs are stored in memory. To test the system, the choice has been made to reconfigure the Scrambler, Interleaver, FEC, and IFFT blocks in order to perform vertical handover during runtime. The configuration controller reconfigures these modules according to the VHA requests.

## 3.5 Experimental Results

In this section, the experimental results for the comparison between the two proposed reconfigurable architectures are shown and discussed. Also, we present the performance of the unified chain for WiFi and WiMax in terms of reconfiguration time, FPGA resources, and power on chip.

### 3.5.1 Performance Evaluation for MRBA and ORBA

In this section, the two reconfigurable architectures are used to implement a unified reconfigurable chain for WiFi-WiMax standards. To evaluate and test both schemes, partial reconfiguration is applied on an OFDM chain implemented on the ZedBoard. Blocks like Interleaver, Scrambler, and Reed-Solomon Encoder are added to the OFDM chain and others are modified in order to make it similar to the physical layer chain of the studied wireless standards. As mentioned earlier, to use MRBA, the blocks of the different chains are compared to determine the similarities and differences. Similar blocks with the same interface and functionality are defined as static modules in the design. Blocks with the same interface but different functionalities are defined as RMs in the design. Using MRBA for the unified chain of WiFi and WiMax the blocks defined as reconfigurable blocks are : TX Controller, Interleaver, Encoder, Scrambler, Modulator, and IFFT. The static part is formed by the remaining blocks that are used for both standards without reconfiguration. The partial bit-streams for the different versions of the reconfigurable blocks are generated and stored in memory. When using ORBA for switching between WiFi and WiMax, all the chain is considered as one reconfigurable block. In this case, only two partial bit-streams for both systems are generated.

During the partial reconfiguration process, the data flow is suspended and data are saved in a buffer until the PR operation is done. It is important to know the time needed to reconfigure the system so we can compute the minimum size of the buffer. The time needed to achieve the partial reconfiguration in both architectures is computed by Equation. 3.1 (MRBA) and Equation. 3.2 (ORBA).

$$T = \sum_{i=1}^N P_i + \sum_j^M R_j \quad (3.1)$$

$$T = \sum_{i=1}^N P_i + R_T \quad (3.2)$$

Where  $N$  is the number of all the blocks (static and reconfigurable) in the chain.  $P_i$  presents the processing time needed by each block to process the input data and generate the output data.  $M$  represents the number of RMs in the chain.  $R$  is the reconfiguration time needed for each module.  $R_T$  represents the reconfiguration time of the full chain in case of the ORBA scenario. In fact, according to the equations we can say that it is best to use ORBA when  $Rt < \sum_j^M Rj$ .

If the equations are applied in the case of WiFi-WiMax unified chain, we will get the following in MRBA scenario :

$$\begin{aligned} T = & (P(\text{scrambler}) + P(\text{Encoder}) + P(\text{Interleaver}) + P(\text{Modulator}) + P(\text{Mapper}) \\ & + P(\text{IFFT} + cp) + P(\text{Pilotguard}) + P(\text{RF})) + R(\text{scrambler}) \\ & + R(\text{Encoder}) + R(\text{Interleaver}) + R(\text{IFFT}). \end{aligned}$$

For the ORBA scenario the equation is :

$$\begin{aligned} T = & (P(\text{scrambler}) + P(\text{Encoder}) + P(\text{Interleaver}) + P(\text{Modulator}) + P(\text{Mapper}) \\ & + P(\text{IFFT} + cp) + P(\text{Pilotguard}) + P(\text{RF})) + R(\text{UnifiedChain}). \end{aligned}$$

To create partial bit-streams, partial reconfiguration has been applied to the unified chain using Vivado tools. During the floorplaning stage, each reconfigurable block is enclosed in a reconfigurable region on the FPGA. The size of the region depends on the resources needed by the block. The size of the output partial bit-stream is directly related to the size of the reconfigurable region. The smallest partial bit-stream that can be generated has a size of 30 kB. TABLE 3.2 presents the partial bit-streams size and the reconfiguration time needed for the reconfigurable blocks in case of the MRBA and for the full chain in case of the ORBA. The partial bit-stream size is related to the complexity of the generated reconfigurable block. The results show that, in the WiFi-WiMax scenario, the reconfiguration time needed in the case of MRBA is :

$$\begin{aligned} T = & \text{rec\_time\_TX\_Controller} + \text{rec\_time\_Interleaver} \\ & + \text{rec\_time\_Encoder} + \text{rec\_time\_Scrambler} \\ & + \text{rec\_time\_Modulator} + \text{rec\_time\_IFFT} \\ = & 0.238 + 0.234 + 0.316 + 0.231 + 0.241 + 1.41 = 2.67ms. \end{aligned}$$

Alternatively, the time needed to reconfigure the full chain is 4.45 ms. TABLE 3.2 shows the difference in terms of the required reconfiguration time between full and

partial reconfiguration. The full chain reconfigurable module will include all the static blocks in addition to the reconfigurable one (fifos, wrappers, and similar blocks). Using MRBA for implementing WiFi-WiMax reconfigurable unified chain gives the best reconfiguration time by approximately 1/2 compared to ORBA design. These results are not fixed, and they are actually related to the considered scenario. In most cases, MRBA provides the best reconfiguration time compared to ORBA design, but the benefit is related to the number of reconfigurable blocks in the unified chain. The time needed to update the partial bit-streams depends on their size and the available throughput. Since in ORBA the partial bit-streams size is relatively large, then the time needed to update a partial bit-stream is greater than that in the case of MRBA. In addition, in ORBA for example, if the system needs to update the Encoder block (40 kB), the partial bit-stream of all the chain (317 kB) is updated. On the other hand, in MRBA, it is sufficient to update only the Encoder block.

### 3.5.2 Experimental Results for the Unified Chain for WiFi-WiMax

The experimental results of the WiFi-WiMax vertical handover scenario are presented in this section. Vivado power and utilization estimation tools had been used to estimate the power consumption and the reserved resources of the different versions of the RMs.

Module	Partial Bit-Stream Size	Reconfiguration Time
TX Controller	30 KB	0.238 ms
Interleaver	30 KB	0.234 ms
Encoder	40 KB	0.316 ms
Scrambler	30 KB	0.231 ms
Modulator	30 KB	0.241 ms
IFFT	180 KB	1.41 ms
One chain	317 KB	4.45 ms
Full reconfiguration	3951 KB	55.46 ms

TABLE 3.2 – Partial bit-streams size and reconfiguration time

TABLE 3.3 – Size and power consumption of the considered blocks

Module	Power (mW)	FPGA Resources (LUT-REG-RAM-DSP)
Scrambler (WiFi)	less than 0.1	10 - 19 - 0 - 0
Scrambler (WiMax)	less than 0.1	12 - 20 - 0 - 0
Interleaver (WiFi)	less than 0.1	68 - 95 - 0 - 0
Interleaver (WiMax)	less than 0.1	75 - 100 - 0 - 0
FEC Encoder (WiFi)	less than 0.1	30 - 40 - 0 - 0
FEC Encoder (WiMax)	0.1	129 - 157 - 1 - 0
64-point IFFT (WiFi)	14	1185 - 1965 - 0.5 - 6
256-point IFFT (WiMax)	20	1676 - 2690 - 1 - 9

TABLE 3.3 shows the resources utilization and the estimated power consumption of each version of the reconfigurable blocks. As a result of applying the partial reconfiguration, the power consumption and the used resources on FPGA are reduced. As an alternative, both WiFi and WiMax standards would be implemented in parallel. This typically requires a lot of resources on FPGA. As a consequence, additional static and dynamic power would be consumed by the chip.

TABLE 3.4 – Comparison of hardware resources usage

	LUTs	REGs	DSPs
unified chain for both standards	2229	3376	9
both standards implemented	3967	6027	15
Available on ZYNQ	53200	106460	220

TABLE 3.4 shows the amount of used resources in two cases : first, when a unified chain is implemented for both standards, and second, when they are implemented in parallel. For the transmitter, the results show that the used resources are reduced approximately by a factor of 1.7, when the partially reconfigured unified chain is used. The percentages of used resources when both standards are implemented is 6.9%, 5.6%, and 6.81% for LUTs, REGs and DSPs respectively, on

the other hand the percentages are 4.18%, 3.17%, and 4.09% when a unified chain is used for both standards.

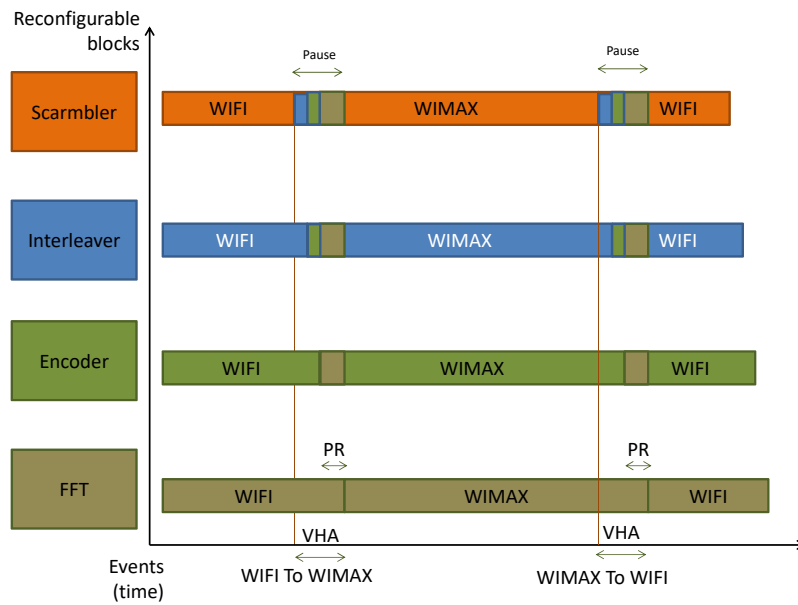


FIGURE 3.5 – State of the RMs during VHA

In FIGURE 3.5, a Gantt chart illustrates a scenario that shows the state of the RMs implemented on FPGA when VHA is applied. The chart is drawn according to the measured reconfiguration times and reflects how the unified chain switches from one standard to another. As shown in the figure, the modules are reconfigured sequentially in an order that is related to their position in the chain. The time needed to achieve switching from one standard to another is the sum of the reconfiguration times required by each module.

As noticed, when a module is reconfigured, the data flow is paused until the subsequent modules are also reconfigured. As mentioned earlier, the time needed to reconfigure a block is related to its size. This is illustrated in FIGURE 3.5 in which the time needed to reconfigure the IFFT block is greater than that required by the other ones. The switching from WiFi to WiMax is achieved when the last module (for ex. IFFT) is reconfigured successfully. Finally, data are transferred

through the WiMax standard until another VHA occurs.

## 3.6 Summary

In this chapter, an ARM-FPGA platform for auto adaptive wireless communication systems has been proposed to switch between multiple wireless standards. The general system design blocks on both PS and PL was presented and explained in the current case. The mechanism of applying the partial reconfiguration technique in a unified reconfigurable chain for multiple wireless communication systems was discussed. Two reconfigurable architectures (ORBA and MRBA) were proposed and compared. The advantages and disadvantages of these architectures were presented. Selecting the best reconfigurable architecture depends on the desired application. MRBA is considered more appropriate for wireless standards with similar PHY layer architecture. On the other hand ORBA is better for wireless standards with different PHY layer architectures. To test the proposed ARM-FPGA system and evaluate the two proposed architectures, a unified chain for WiFi and WiMax standards was considered as a case study. The similarities and differences of the two standards were studied to obtain a reconfigurable unified chain for both standards. The experimental results were extracted after implementing and testing the unified chain on the ZedBoard. It was clearly shown the advantages of using PR technique to reconfigure unified chain for multiple wireless standards.

The switching between the PHY layers of multiple wireless standards is considered as the execution of the decision made by VHA. It is very important for the VHA to be smart enough and understand the environment to provide an efficient decision. This will be discussed in details in the next chapter.





## CHAPTER 4

# INTELLIGENT VHA FOR WIRELESS NETWORKS USING NEURAL NETWORKS AND REINFORCEMENT LEARNING

### 4.1 Introduction

In the previous chapter the implementation of a unified reconfigurable PHY layer for multiple wireless standards on ARM-FPGA platform has been discussed. PR technique was adopted to implement the reconfigurable system. Besides, it is important to study the algorithms that make the decisions to perform the reconfiguration operation. In our system, they will run on the ARM and request partial reconfiguration for the PHY chain. In this chapter, we will discuss the concepts of the VHAs in wireless systems and propose an intelligent VHA based on NN combined with reinforcement learning. In the next section, we will present how VHAs work.

### 4.2 Heterogeneous Networks and Vertical Handover

The wireless network formed of multiple wireless standards is called heterogeneous network. VHAs will have the task to sense the wireless spectrum conti-

uously and detect the available wireless standards.

VHAs read different parameters to learn, study, and then decide whether to perform a vertical handover or keep on communicating through the current standard. The key of taking a good decision is driven by the collected parameters from the environment. The known parameters can be divided into four categories :

1. **Wireless Channel Parameters** : Reading parameters from the wireless channel to inform the VHA about the channel status. RSS, SNR, and BER are examples of such parameters.
2. **Network Information Parameters** : These parameters inform the VHA about the network status. This category includes the number of connected nodes to the access point, data rate, delay, jitter, and cost.
3. **User Parameters** : These parameters reflect user's data requirements at the application layer such as the required data rate and high priority data.
4. **System Parameters** : Such type of parameters provides the algorithm with the device status. These parameters may deal with power consumption, battery level, and device speed.

### 4.3 VHA for WiFi and WiMax Heterogeneous Network

In our work we have considered heterogeneous networks are composed of WiFi and WiMax. Each radio access technology features different specifications in terms of supported Uplink and DownLink data rate and coverage range. Both WiFi and WiMax standards support the IP protocol that is widely used nowadays for voice, data, and video streaming. The comparison between the two wireless standards is presented in TABLE 4.1.

The proposed VHA is considered as the decision agent in the proposed ARM-FPGA platform. The ARM-FPGA based end-node features a custom VHA running on the PS part and an unified partially reconfigurable WiMax-WiFi physical layer chain implemented in the FPGA.

TABLE 4.1 – WiMax and WiFi networks comparison

Characteristic	WiMax	WiFi
Range	up to 40 Km	up to 100 meters
Scalability	user scale one to hundred	user scale from one to ten
Bit Rate	up to 100 Mbps in 20 MHz channel	up to 54 Mbps in 20 MHz channel
QoS	provide several level of QoS	Does not guarantee QoS service
Mobility	support mobility	fixed

### 4.3.1 Vertical Handover Algorithm based on Adaptive Scoring System

The proposed VHA for an WiMax–WiFi network is composed of three stages : an Handover Trigger, an Initial Decision and a Final Decision. The algorithm identifies two execution contexts. In the first context, the system communicates through the WiMax standard, whereas, in the second one the system is initially operating with WiFi.

The handover algorithm that switches from WiMax to WiFi is illustrated in FIGURE 4.1. The trigger is initiated as soon as the handover process detects that the RSS of WiFi is greater than that of WiMax. The trigger is based on RSS because it is considered as the simplest and fastest way to detect the availability of wireless standards. The Initial Decision state is used to make fast decisions and detect fake handover. Fake handover is detected if the speed of the device is greater than a certain limit defined by `WIFI_max_speed_limit` (similar to that defined in [JH09]). In this case, if the VHA is only based on RSS, the system will switch to WiFi. Since the node is in a high speed it will leave the WiFi Access Point (AP) and the system will switch back to WiMax. In this case, two handovers would have occurred without any benefit, in addition to the waste of time and power. Thus the system is able to detect and avoid type of fake handover.

If the Initial Decision state has successfully passed, the system goes into the Final Decision state. In this final state, an adaptive scoring system reads multiple parameters to decide which network is the best according to the user preferences. If the scoring system decides that WiFi is better than WiMax, then vertical handover occurs. The handover algorithm from WiFi to WiMax is illustrated in FIGURE 4.2. Since the initial state is WiFi, the trigger is initiated when the handover process detects that the RSS of WiMax is greater than that of WiFi. The Initial Decision state explores two conditions. The first condition consists in checking if the battery

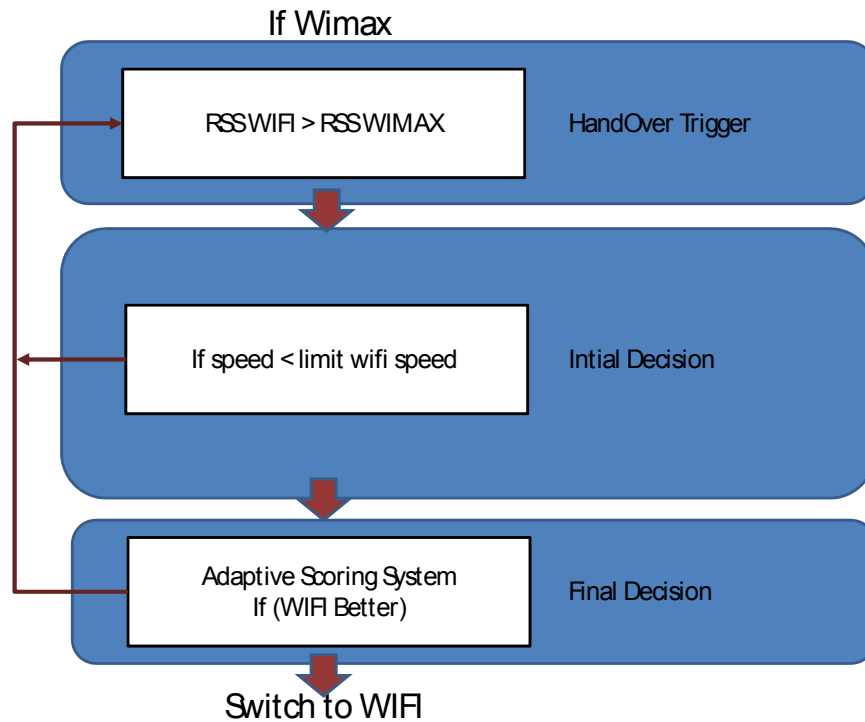


FIGURE 4.1 – From WiMax to WiFi Vertical Handover Algorithm

level is less than a certain limit defined by `battery_low_level_limit`. The second condition makes sure that the RSS of WiFi is less than a certain limit defined by `RSS_WIFI_limit`. These two conditions should be verified, otherwise the handover is canceled. This initial condition will prevent another type of fake handover where the battery level is low but the signal received from WiMax is better than that from WiFi. Although WiFi is still available and the power permits communicating through WiFi only. So if the decision is based on RSS only the system decides to switch to WiMax and the power consumption increases. In this case if the battery level is critical the system will power off soon, while on the other hand the Initial Decision detect this situation and neglect the trigger.

If the Initial Decision state is successfully passed, the system reaches the Final Decision state. At this state, an adaptive scoring system reads multiple parameters to decide which network is suitable according to the user application processes preferences. If the scoring system decides that WiMax is better than WiFi, then vertical handover occurs.

The general concept of the scoring system is based on assigning values to each selected parameter in the VHA decision. Values assigned to parameters are added

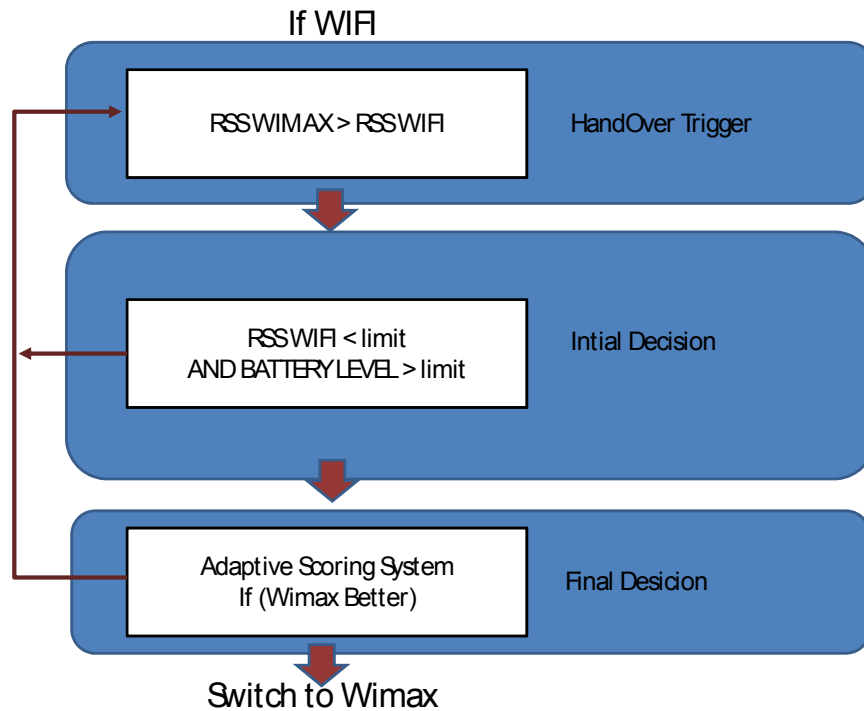


FIGURE 4.2 – From WiFi to WiMax Vertical Handover Algorithm

to the score of the best wireless standard according to each parameter. The wireless standard with the highest score wins and the VHA decides to switch to it. An example dealing with the operation of the proposed scoring system is shown in TABLE 4.2, where each parameter is assigned different points. The scoring system operates as follows : If a standard provides a better performance regarding a given parameter (ex. SNR, BER), then the points related to this concerned parameter are added to the score of the corresponding network. For example, if the Data Rate parameter is assigned 3 points, and WiFi provides better throughput than WiMax, then 3 points are added to the WiFi score. On the other hand, if the delay value in WiMax is less than that in WiFi, 4 points are added to WiMax. Finally, the network with the highest score is selected. In this example, the values assigned to the parameters are static. To increase the performance of the system adaptivity is added. In this case, the values assigned to the parameters are changed according to specific conditions.

Adding adaptivity to the scoring system improves the efficiency of the system. For example, we consider the user preferences parameter to be the main parameter which affects the score of the other parameters. In this case, it will be possible to

TABLE 4.2 – Example of a scoring system

<b>Parameter</b>	<b>WiFi</b>	<b>WiMax</b>
Data Rate (3)	<b>3</b>	-
Delay (4)	-	<b>4</b>
Jitter (3)	-	<b>3</b>
SNR (3)	-	<b>3</b>
BER (3)	-	<b>3</b>
Power Consumption (5)	<b>5</b>	-
<b>Total Score</b>	<b>8</b>	<b>13</b>

adapt the weight of each parameter in the scoring system according to the type of application running in the user space. As shown in TABLE 4.3, the points attributed to a given parameter may change according to the type of the running application . The applications can be divided into four categories. The first category includes the video applications that demand high data rate and low delay and jitter. The second category is the VoIP applications. This type of applications requires less data rate than Video applications but similar delay and jitter specifications. The third category is the browsing which does not require real time services with average data rate. The fourth case, the ideal case, is when the node is only connected to the wireless network but without running any application. For example, when running a VoIP application, the required data rate, delay and jitter parameters have more weights than in case of running a browsing application [CFY04].

The same concept can also be applied to the power consumption parameter. In this case, the weight of this parameter is related to the current power level in the system. For example the weight of power consumption increases as the battery level decreases. Therefore, selecting the best network is not only related to the user applications requirements but also to the power status of the system. In this case the standard providing less power consumption will have a bigger weight when the power level decreases.

The general equation for the proposed VHA in case of switching from WiMax to WiFi and from WiFi to WiMax is as follows :

$$D(t) = D_I(t, speed, power) * I_M(ScoreR_i(t)) \quad (4.1)$$

where  $R_i$  is selected from set of available wireless systems WiFi, WiMax, ...

$I_M$  select the wireless standard with the highest score at time  $t$ , The initial decision  $D_I$  is applied if the algorithm trigger is true. Otherwise, the system applies the adaptive scoring system. The equation of the initial decision is based on speed and the power level in the end node. Initial Decision gives zero if speed is greater than limit speed of the current standard. This in case of the system is initially using WiMax. The initial decision gives zero also if the system initially using WiFi, and the RSS of WiMax is detected greater than the RSS of WiFi but the power level is low. So, the possible values for  $D_I$  is 0 or 1.

The equation of the adaptive scoring system :

$$ScoreR_i(t) = \sum_{j=0}^M aW(x_j, t) \quad (4.2)$$

where  $a = 1$  if wireless standard has best weight for parameter  $x$  at time  $t$  else 0  
 $X_j$  parameters delay , network load, throughput, SNR, BER, power consumption ...  
 Score of a standard is the sum of each parameter score at time  $t$ .  $W(x_j, t)$  weight of parameter  $x_j$  at time  $t$  related to the category of the parameter. The equations for the adaptive weight for QoS and power consumption :

$$QoS : w(t) = user\_pref$$

$$Power\ Consumption : w(t) = (100 - battery\_level(t))/20$$

As noticed in the above equations, the QoS weight is related to the user preferences value. The user preferences value depends on the running applications, for example 4 for video, 3 fo VoIP, 2 for browsing and 1 for Ideal state. In this case the weight of the QoS parameter will change as the user runs different applications. In the second equation we notice how the power consumption parameter varies as the power level in the system varies. As the power level in the system decreases the weight of the power consumption increases.

The proposed VHA based on adaptive scoring system can be improved by adding the machine learning technique. In the next section, an VHA based on NN combined with reinforcement learning is proposed and analyzed.



TABLE 4.3 – Scoring system example according to different applications

Parameter	VOIP	Video	Browsing	Ideal
Data Rate	3	4	2	1
Delay	4	4	2	1
Jitter	4	4	1	1

## 4.4 Proposed VHA based on Machine Learning

In the previous section, we have presented an VHA for WiFi-WiMax heterogeneous network based on fast initial decision and adaptive scoring system. It is interesting to use the Machine Learning (ML) techniques to make the system more intelligent.

### 4.4.1 Proposed System Design

In this work, we propose a system that combines both supervised and unsupervised machine learning algorithms to switch between different standards. This system is implemented in the End-Device. It is based on an improvement of the previously proposed adaptive scoring system. The main idea is to benefit from this previous work in a first step. This step consists in collecting records and rewards from the previous adaptive scoring system that is described in Section 4.3. The taken decisions are then classified based on the obtained rewards. In a second step, only those generating good rewards are considered and they will form a new data set. The collected records and corresponding acceptable or good rewards will be used to train the NN system.

At a certain time and after collecting a large data set, the NN system is considered as sufficiently trained to take a right decision. Thus, at this point, it may be adopted by the system to take further decisions. The system can repeat the procedure of training the NN system after updating the data set in order to optimize its operation and improve its behavior. The proposed system consists of the blocks shown in FIGURE 4.3.

#### 4.4.1.1 Parameters Provider

This block collects all needed parameters from different layers (physical, MAC, Network) to get information about the status of the channel, networks, user run-

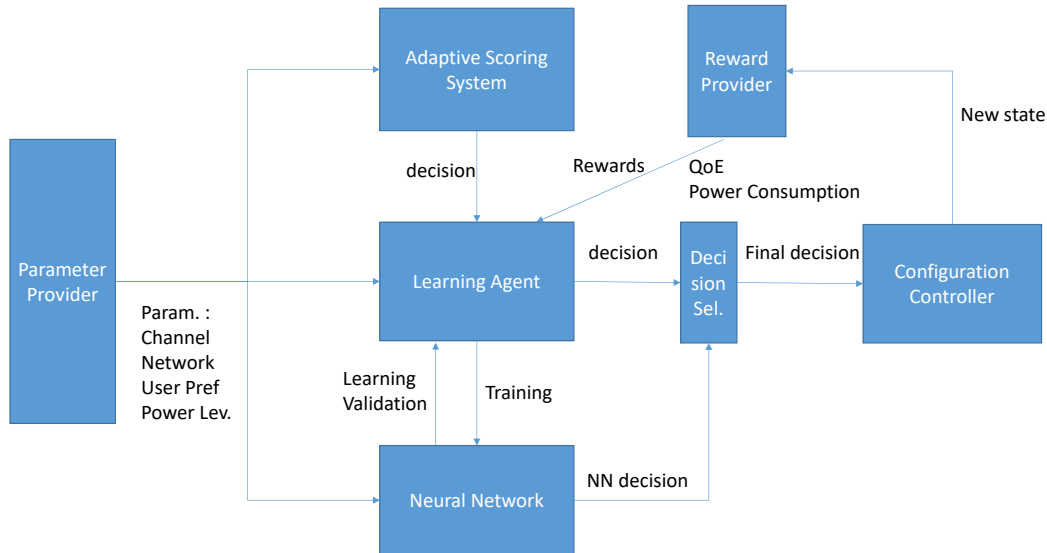


FIGURE 4.3 – Proposed VHA design

ning applications, and also node status (location, power, speed). It is considered as an interface for all the parameters of the system. The input of this block is the parameters from different layers.

#### 4.4.1.2 The Neural Network

In this study, we consider basic MLP feed forward networks. They permit to model complex behaviors and allow performing multi-dimensional functions approximation. Such networks have one or more hidden layers composed of neurons with non linear transfer function. Also they provide an output layer that implements output neurons with a linear activation function. In a first learning phase, the multiple layers with nonlinear activation functions allow the network to learn the relationships between inputs and outputs. This is performed by modifying the weights values between the different neurons. In a second phase (the forward phase), the network may estimate the correct output for any given input pattern.

In our models, three layers have been used. Each layer receives its inputs from

the precedent layer and forwards its outputs to the subsequent one. In the forward phase, the weight matrix of the hidden layer is multiplied by the input vector  $X = (x_1, x_2, x_3, \dots, x_n)^T$ , to compute the hidden layer output, as expressed in equation 4.3.

$$y_{h,j} = f \left( \sum_{i=1}^{N_i} w_{h,ji} x_i - \theta \right) \quad (4.3)$$

where  $w_{h,ji}$  is the weight connecting input  $i$  to unit  $j$  in the hidden neuron layer.  $\theta$  is an offset termed as bias that is also connected to each neuron. In order to train the networks, the back-propagation algorithm has been used.

The inputs of this block are the parameters, decision, status, and reward for each iteration during the training phase and the output is the learning validation. When training is validated, the output is the NN decision.

#### 4.4.1.3 Adaptive Scoring System

At the beginning of this chapter, we have proposed a VHA based on adaptive scoring system for WiFi-WiMax networks. The proposed VHA is composed of an initial decision and a final one. The initial decision is a direct function based on the speed and power level of the node. The final decision is the adaptive scoring system.

The role of the adaptive scoring system is to make decisions at initial time where NN is not trained yet. These decisions are saved as data records to train the NN function later. The reward is computed after each decision, and data with good rewards are tagged while data with bad rewards are ignored. Only tagged data are used to train the NN.

The inputs of this block are the parameters from the parameter provider and the output is the decision of the adaptive scoring system.

#### 4.4.1.4 Learning Agent

This module has an important role in the system. Its goal is to train the NN function with the best records collected from the adaptive scoring system. As mentioned above, initially and while the NN is not trained yet, the adaptive scoring system will make decisions. The records are formed by parameters and decisions at time  $t$  and rewards at time  $t + 1$ . The best records are tagged based on the received rewards, and they are used to train the NN function. The learning agent

changes the decision made by the adaptive scoring system for similar parameters to get a different reward at time  $t + 1$ . In this case, the learning agent will tag the record with the best reward to use it later in the training of the NN. The role of the learning agent can be summarized by the following points :

- Collect and save records : parameters, control signals, decisions, and rewards.
- Invert the decision for a given record with the same parameters and control signals to get a new record.
- Select the best records to provide a new input for the NN training.
- Train the NN during runtime when new records are available.
- Validate and test the results from the NN to use them only when the NN provides efficient decisions.

The inputs of the learning agent block are the parameters, decision from adaptive scoring system, rewards from reward provider, and training validation from NN block. The learning agent train the NN blocks by tagged records and forward adaptive scoring system decision to the decision selector.

#### 4.4.1.5 Reward Provider

The reward provider provides the learning agent with the rewards after taking a decision to switch from a state to another. For example, we consider a system including the QoS and power consumption as input parameters. The system can study the effect of any decision based on the QoS and power consumption. The reward can be a numerical value for power consumption and QoS. The QoS values reflect the quality of communication (throughput, delay, jitter) that must be achieved for the running user application. For example the power consumption reward is measured after making a decision to switch from a standard to another, if the power consumption decreases the reward is considered as a good reward whereas if the power consumption increases the reward considered as bad.

#### 4.4.1.6 Decision Selector

This block receives the decision from the NN and learning agent. When the NN module is validated by the learning agent, the decision agent forwards the decisions made by the NN module to the configuration controller and the other modules are suspended. At this level, the system should have learned to make good decisions based on the experience and using machine learning algorithms. The NN module

is validated when it is sufficiently trained by the minimum required number of tagged records. This limit is selected based on experimental results.

The inputs of this block is the decisions from learning agent and from NN, the output is final decision which is forwarded to the configuration controller.

#### **4.4.1.7 The Configuration Controller**

The goal of the configuration controller consists in transferring partial bit-streams to the FPGA. For additional security reasons, the partial bit-streams memory locations are only accessible from the configuration controller. This controller eliminates any conflict that may occur when several user processes decide to access the FPGA fabric at the same time. In this case, it transfers the bit-streams consecutively.

#### **4.4.2 NN based VHA Vs. standard VHA**

Standard VHA are proposed to improve the performance of a system. For example in terms of power consumption or QoS. Sometimes, the VHAs aim to improve the performance of the end node only, while other types of VHAs aims to improve the performance of the whole system. In the standard VHAs, the process is static and depends only on the current status of the system. These static VHAs can be described by mathematical models. The results obtained from the static VHAs are not considered as optimal solutions and in some cases they provide bad results. On the other hand, NN based VHAs analyze previous data to provide a non linear black box function. All previous data (inputs and outputs) are stored and used to create the NN box. The input parameters are always changing due to the mobility of the user, the existence of multiple wireless standards, and the characteristics of the wireless channel. Thus, it is difficult to create a relation between inputs and outputs using mathematical or statistical models. On the other NN models can do that.

The main advantages of using NN are :

- Can handle large amount of data sets ;
- Has the ability to implicitly detect complex nonlinear relationships between dependent and independent variables ;
- Has the ability to detect all possible interactions between predictor variables.
- It is considered very suitable for inverse modeling when the numerical re-

lations between input and output variables are unknown, and cannot be established.

Another major advantage of using NN over other statistical techniques is its capability to capture non linear relationship among concerned variables. Sometimes NNs are better than other techniques. Some others will perform worse. It depends on the learning data set. Large data set gives better results. Another disadvantage of NN is that it is a black box function.

## 4.5 Experimental Results

To prove the feasibility of our approach, the proposed system was tested and simulated to obtain experimental results. This gives an idea about the performance of the system.

A simulation was performed based on a WiFi-WiMax scenario using a Java simulator. Based on the adaptive scoring system and a simple reward system, the simulator has been setup to generate the required data (Input Parameters – Decisions – Rewards – and Learning Decisions). The Java simulator randomly generates the input parameters : RSS, Network Load, User Preferences, Power Level and Status. Then, the decisions and related rewards for power and QoS can be selected according to a simple decision and a reward engine. The engine operates based on the input parameters and previous decisions. The records with good and bad rewards are tagged with different values to distinguish between them during training process. The generated data are mapped to numerical results and extracted to a file that could be used later in Matlab. The file contains typical values :

- RSS : High 2, Med 1, Low 0
- Network Load : High 2, Med 1, Low 0
- User Preferences : Video 3, Voip 2, Browsing 1, Idle 0
- Power Level : High 2, Med 1, Low 0
- Status : WiMax 0, WiFi 1
- Decision : No Handover 0, Handover 1
- Reward Power : Decreased 2, Increased 1, Stable 0
- Reward QoS : Increased 2, Decreased 1, Stable 0
- Learn NN : Yes 1, No 0

For example, the sequence *0-1-2-2-2-0-1-1-2-2-1* represents : WiFi\_RSS : low - WiMax\_RSS : med -WiFi\_Network Load : high -WiMax\_Network Load : high

-User Preferences : Voip -Power Level : low -Status : Wifi -Decision : handover  
 -Reward\_Power Consumption : decreased -Reward\_QoS : increased -Learn\_NN :  
 Yes.

The generated data from the Java simulator are used as training data to create a NN function using the Matlab Neural Net Fitting GUI interface. In this Matlab toolbox we have the ability to select the input and output vectors, and the number of hidden layers (in our case 2 layers have selected). Also we can select the training algorithm (ex : Levenberg-Marquardt, and Bayesian Regularization). The result of the NN toolbox is a matlab function with the generated parameters for the NN layers. The generated NN has 7 inputs (RSS\_WiFi, RSS\_WiMax, NI\_WiFi, NI\_WiMax, user preferences, power\_level, status) and has one output (Handover/No Handover) as a real value between 0 and 1.

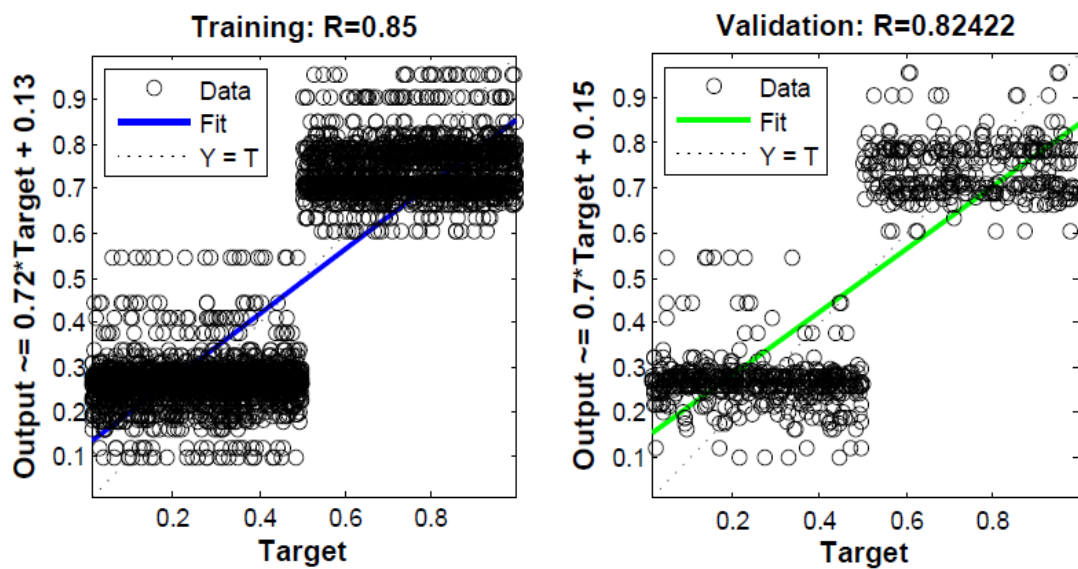


FIGURE 4.4 – Training and Validation System Response of the Neural Network System.

As we have discussed before that the learning agent should know when to train the NN system, also, it has the task to retrain the NN every time there are enough

TABLE 4.4 – Example showing how the learning agent selects the best records

WiFi-RSS	WiMax-RSS	WiFi-NL	WiMax-NL	User pref	PWL	State/ Decision	Rew-PWC	Rew-QoS	Learn NN
High	Med	High	low	Voip	med	WiFi/ Hand.	Increased	Increased	Yes
High	Med	High	low	Voip	med	WiFi/ No-Hand.	—————	—————	No
High	low	High	High	Browsing	low	WiMax/ No-Hand.	—————	—————	No
High	low	High	High	Browsing	low	WiMax /Hand.	Increased	—————	Yes

NL : Network Load ; UP : User Preference ; PWL : Power Level ; PWC : Power Consumption ; Rew : Reward.



TABLE 4.5 – MSE of Training and Validation for different number of samples.

Number of Samples	MSE Training	MSE Validation
175	3.9 e-2	3.8 e-2
688	2.7 e-2	2.9 e-2
5500	2.04 e-2	2.03 e-2

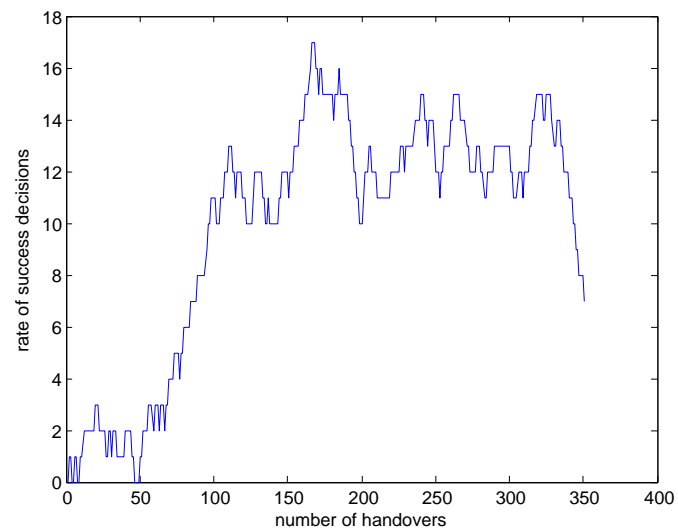


FIGURE 4.5 – Rate of Successful Decisions using all records for Power Consumption.

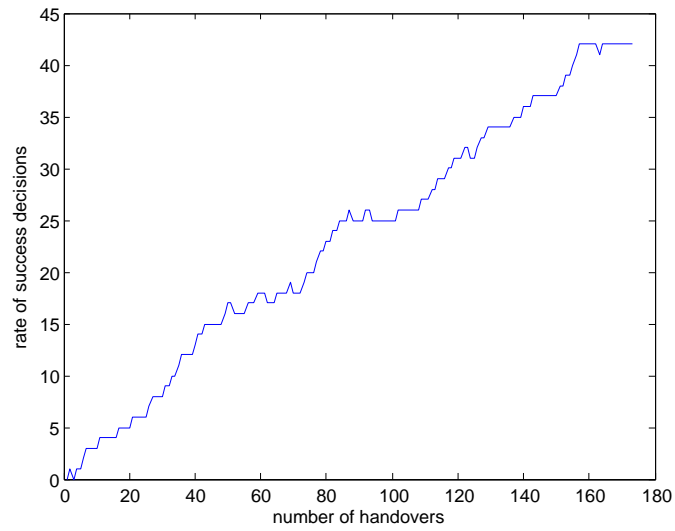


FIGURE 4.6 – Rate of Successful Decisions using tagged records only for Power Consumption.

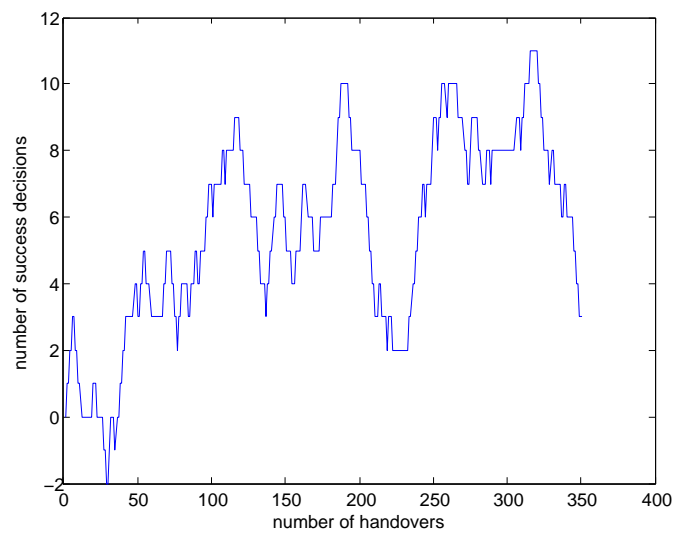


FIGURE 4.7 – Rate of Successful Decisions using all records (QoS)

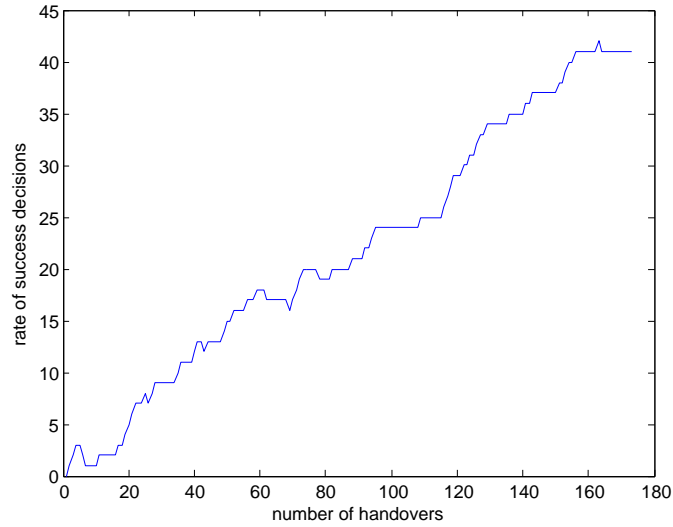


FIGURE 4.8 – Rate of Successful Decisions using tagged records only (QoS)

tagged records. It was important to know what is the best number of tagged records needed to provide good results. In TABLE 4.5, we present the Mean Squared Error (MSE) for different number of records used to train the NN system. As noticed, the more the number of used samples is considered, the better the obtained MSE becomes. Selecting large number of samples will give better results but more time is required to collect these samples.

In FIGURE 4.4, we show the system response (NN decision) while training and validating data of the NN system. It is noticed how the data are divided into two categories (in our case, handover or no handover). Regression R values represent the correlation between the outputs and the targets.

The proposed system is tested in terms of power consumption and QoS. In FIGURE 4.5, we show the rate of successful decisions with respect to the number of handovers using all the records (the tagged and not untagged ones). In FIGURE 4.6, we present the rate of successful decisions with respect to the number of handovers using the tagged records only.

The rate of successful decisions leads to adding one for a good decision and subtracting one for a bad decision to the total sum of decisions. The results take into consideration the Power Consumption (PWC) reward to decide whether the obtained decisions are good or bad. It is clearly noticed that training the NN with

the tagged records that represent the records with only good rewards provides better results.

The same scenario was applied by taking into consideration the QoS reward as the main reward to decide on the quality of the made decision. The results are shown in FIGURE 4.7 for all the records and in FIGURE 4.8 for the tagged records only. The results also show how training the NN with the good records only provides better results.

## 4.6 Summary

In this chapter, we have proposed a VHA for wireless standards based on adaptive scoring system with initial fast decision. The three stages of the proposed VHA are described with an explanation for the adaptivity added to the scoring system. Then, we have proposed an NN based VHA to switch between multiple wireless standards in an heterogeneous network. The NN is combined with reinforcement learning to achieve better performance. A learning agent has been proposed to train the NN system with best records only in order to improve the NN decision. These good records are tagged based on the obtained reward after each taken decision. Experimental results have been presented based on a WiFi-WiMax scenario. The results show that the learning methodology with the use of reinforcement technique provides better performance in the NN system.

After proposing an ARM-FPGA platform for reconfigurable wireless systems using the PR technique, it is important to perform power consumption and real-time measurements for the PR operation and power analysis for the PR design on PL. This will be the subject of the next chapter. In addition, the chapter will contain an explanation for two use-cases in context of PR applications.



# CHAPTER 5

## IMPLEMENTATION AND REAL TIME POWER MEASUREMENTS

### 5.1 Introduction

In previous chapters, we had explained about benefiting from PR technique in adaptive reconfigurable wireless systems and switching between multiple wireless standards. We had also described VHA to decide when to switch between the wireless standards. In this chapter we will go deeply into the PR implementation of the adaptive modulation technique in an OFDM chain. Then we will present the on chip power consumption results and analysis for PR design compared to parallel one. It is important to measure the power overhead added during the PR operation on both PS and PL.

### 5.2 Implementation and Power Analysis for Automated Adapted Reconfigurable OFDM Transmitter

In this section, we present power analysis when using the PR technique in a adapted reconfigurable OFDM transmitter. First, we describe the implementation of the OFDM transmitter on the FPGA, and then we present the PR design for a reconfigurable OFDM transmitter on ARM-FPGA SoC. Finally, we analyze the on-chip power for the PR design compared to parallel design.

### 5.2.1 Implementation of the OFDM Transmitter on FPGA

The generic chain of the SISO OFDM transmitter is formed by different entities. These entities are implemented using VHDL. Some of these entities are created using the Xilinx IP Generator. They are connected by port mapping to form the full chain of the SISO OFDM transmitter.

The first block of the OFDM chain is the turbo encoder, created with the Xilinx IP generator. Its task consists in encoding the transmitted data to help detecting and correcting the corrupted bits at the receiver side. For each input bit, the turbo encoder generates two parity bits concatenated with the original bit. Therefore, its rate is  $1/3$ .

The second block is the modulation block in which the output of the encoder is connected to the input of the modulation block. It generates complex symbols with real and imaginary values. In this section, we will explain how to apply the partial reconfiguration concept on this specific block. The third block is the carrier mapper which consists in mapping the generated symbols to the sub-carriers. The IFFT is another IP core used from Xilinx. It is considered as the main block in the OFDM chain. The IFFT block transforms symbols from the frequency domain to the time domain. The generated time domain signal contains (in addition to the multi orthogonal subcarriers) the cyclic prefix which is transmitted during the Guard Interval (GI) to make the OFDM symbol robust against ISI. In our scenario, we use a 256-point IFFT block.

The blocks that have been previously described can be considered as the main blocks of the SISO OFDM transmitter. The chain contains additional blocks such as the Digital to Analog Converter (DAC) Scaler. This block aims at scaling the resolution of the IFFT output. After this step, the signal is ready for digital to analog conversion and is then passed to the Radio Frequency (RF) block that transmits information through the channel. A module for time pilot symbol insertion is also added to insert a pilot at the beginning of each frame for channel estimation at the receiver side. A global controller is also needed to enable, reset and pause all the modules. It also synchronizes the data flow between the blocks. This module is based on a state machine model. It generates an OFDM frame that contains a time pilot signal and ten OFDM symbols. An initialized ROM (Read Only Memory) is used to generate the input data, and also to hold the known time pilot samples. FIFO (First In First Out) blocks are also implemented to adapt the rate between the different modules of the system.

The VHDL code of the SISO OFDM transmitter has been implemented using

the Xilinx ISE design suite and tested with the Modelsim Simulator. The results generated by the Modelsim simulator have been compared to those obtained using Matlab to ensure and validate the functionality of the whole system.

### 5.2.2 Applying Partial Reconfiguration on OFDM Transmitter

The parallel implementation of the adaptive modulation technique of an OFDM chain is shown in FIGURE 5.1. This is considered as a solution to implement it on FPGA and it uses a multiplexer to select the desired modulator. The other solution consists in using the PR technique which will be discussed next.

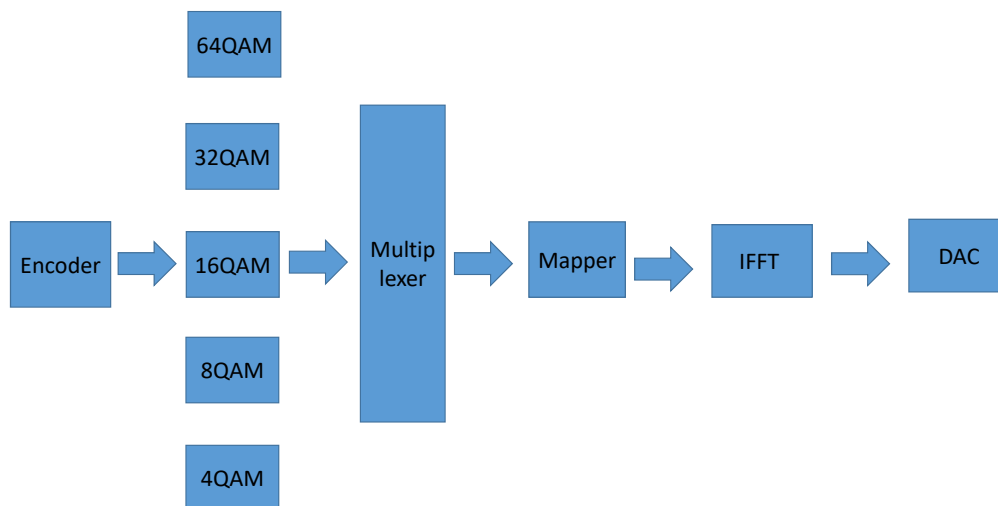


FIGURE 5.1 – Parallel implementation for adaptive modulation in OFDM chain

According to the partial reconfiguration design flow from Xilinx, the partial reconfiguration concept has been implemented in the chain after testing the VHDL code. To implement the adaptive modulation technique, the DPR concept has been applied on the modulation block. In this case, the encoder, mapper, IFFT block, DAC scaler, and FIFO blocks constitute the static part of the system. The modulation block constitutes the dynamic part of the system as shown in FIGURE 5.2. Using Xilinx ISE or Vivado Tools, the modulation block has been resynthesized as a black box. A black box is an implementation of an entity with ports definition but without any architecture. Synthesizing a black box generates



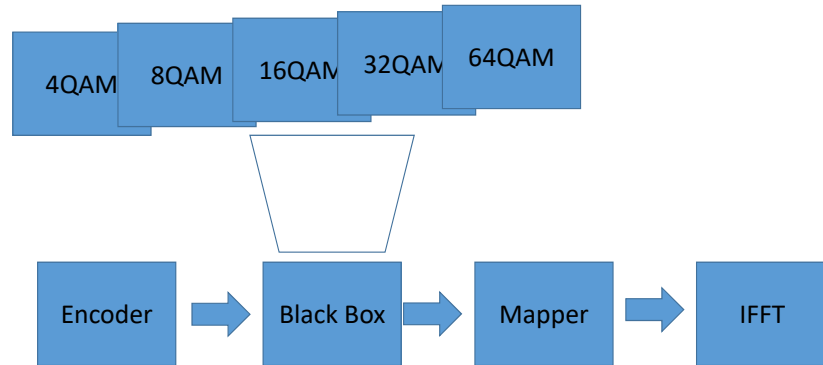


FIGURE 5.2 – PR implementation for adaptive modulation in OFDM chain

a netlist with a constraint information file (ngc). In addition to this netlist file, it is required to import the netlist files of the static modules in order to create a DPR project using Vivado Tools. When creating the DPR project and adding the netlist files, Vivado Tools recognize the black box and select it as a reconfigurable partition. The next step consists in selecting the area of the RM on the FPGA (floorplanning). Then different netlist files of the modulation block are created with different contents but with the same ports definition. In our case, we have created netlists for 4QAM, 8QAM, 16QAM, 32QAM, and 64QAM modulation schemes. All the VHDL entities of the constellation schemes have the same ports definition but different architectures. The corresponding netlist files of the modulation blocks are added to the reconfigurable partition. Finally, we obtain an OFDM chain including a static part and a dynamic part.

Using Vivado Tools, it is possible to generate different bit-streams for programming the FPGA. First, a basic bit-stream is generated, which contains all the static blocks and one configuration corresponding to the reconfigurable block of the system. The main bit-stream is promoted to create different reconfigurable ones containing the different configurations of the reconfigurable partition. The partial bit-streams are very small in size compared to the main bit-stream file. In our case, we have considered a main file with a 4QAM modulation block, and four partial bit-streams for the 8QAM, 16QAM, 32QAM, and 64QAM modulation schemes. To test the partial reconfiguration concept, we programmed the FPGA with the full bit-stream and then using a C code running on the ARM processor we reconfigure it by sending different partial bit-streams. The modulator module is reconfigured while the remaining parts of the system is still running. This was

performed manually, in the next paragraph, we will explain how the system was reconfigured to make it working dynamically.

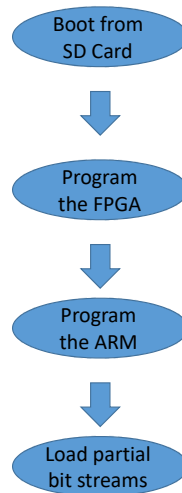


FIGURE 5.3 – Boot sequence on the ZedBoard

### 5.2.3 Automated Adapted Reconfigurable OFDM Transmitter

The system was reconfigured to apply PR on PL from PS automatically. In this case the system will have the ability to reconfigure the reconfigurable hardware modules on PL based on decision taken by processes running on PS. This is performed by configuring the system to reconfigure the PL and PS, load partial bit-streams, and run the application on PS.

During the boot process, the system sends the main bit-stream to the FPGA, and then configures the PS. It also loads the partial bit-streams to the memory allowing the PS to access it during runtime. An illustration of the booting sequence is shown in FIGURE 5.3. For that, we used the first stage boot loader (FSBL) C code provided by Xilinx SDK. FSBL uses the main bit-stream and the main software executable (.elf) file to create a BOOT.BIN file. This file is loaded to the SD card. By configuring the jumpers on the Zedboard, the system automatically boots from the SD card and searches for the BOOT.BIN file. FSBL loads and then gets access to the partial bit-streams stored on the SD card. Initially, it programs

the PL with the main bit-stream. Second, it programs the PS with the main software. Finally, it loads the partial bit-streams into the RAM embedded in the board. The main software part has then the addresses of the partial bit-streams in the memory. While the OFDM chain is running in the PL, the PS applies the adaptive modulation technique. According to the SNR value, the PS decides which modulation must be selected. As a result, the suitable partial bit-stream is transferred to the PL.

#### 5.2.4 On Chip Power Results

After we have created an adapted reconfigurable OFDM transmitter using PR technique, it is important to analyze the on chip power consumption compared to the parallel design.

The power consumed by the FPGA chip is the sum of the static and dynamic power consumptions. The static power is mainly due to the transistor leakage. It occurs even when the transistor is logically "off". The dynamic power is the sum of the transient power consumption and capacitive load power consumption. The transient power consumption represents the amount of power consumed when the device changes its logic state, i.e. logic "0" to logic "1" or vice versa. The capacitive load power consumption represents the power used to charge the load capacitance. There are 3 possible ways to compute the power consumed on FPGA :

- Concept phase : A rough estimate of power can be obtained based on estimates of logic capacity and activity rates.
- Design phase : Power can be calculated more accurately based on detailed information about how the design is implemented in the FPGA.
- System Integration phase : Power is calculated in a lab environment.

In our case, we will use the Xpower Analyzer, a design phase analyzer, to estimate the static and dynamic power consumptions. The power measurements are applied in two scenarios. In the first scenario we implement only the modulator independent from the full OFDM chain. In this case, the modulator is the top level module in the VHDL code. The power results for the two designs, parallel and PR designs are shown in TABLE 5.1 and TABLE 5.2 respectively.

In the second scenario, the adaptive modulator is a module in the OFDM transmitter, the power measurements are retrieved for the modulator module in two cases, the parallel and the PR designs. In the first case, all QAMs are implemented with the controls and the multiplexer. The logic and signal power, which reflects the static and dynamic power respectively, for the Multi-QAM are shown

in TABLE 5.4. In the DPR case only one RM is implemented in the OFDM chain. The results for DPR design are presented in TABLE 5.3. In TABLE 5.5, we present the total on chip power and the dynamic power for all the OFDM transmitter in the two designs.

In the PR design, the number of blocks running at the same time is reduced. Thus, as it can be noticed from the results presented above, this approach results in reducing the power consumption on the chip. The static power consumption is related to the internal architecture of the FPGA and is therefore not affected by the design change. On the other hand, as results show, the dynamic power consumption is reduced when applying the DPR concept. The results in TABLE 5.5 show a 1.6% reduction rate of the total power on chip and a 18.7% reduction rate in dynamic power.

TABLE 5.1 – Power Results for modulator block only using DPR design

Block	Total on chip(mW)	Total no IO's(mW)	Dynamic Power(W)
Control-fifo-mem	0.117	0.115	0.00022
QAM	0.134	0.115	0.00024
Total	0.251	0.230	0.00046

TABLE 5.2 – Power Results for modulator block only using parallel design

Block	Total on chip(mW)	Total no IO's(mW)	Dynamic Power(W)
Multi.Control-fifo	0.127	0.118	0.00164
Multi-QAM	0.146	0.121	0.00345
Total	0.273	0.239	0.00509

TABLE 5.3 – Power Results for QAM and Control blocks in full OFDM chain using DPR design

Block	Logic Power (mW)	Signal Power (mW)	Total (mW)
QAM	0. 01	0. 05	0. 06
Control-fifo mem	0. 02	0. 04	0. 06
Total	0. 03	0. 09	0. 12

In the next section we will show real time power consumption measurements for PR process on PL and PS.

TABLE 5.4 – Power Results for total control and QAM blocks using parallel design

Block	Logic Pow.(mW)	Signal Pow.(mW)	Total(W)
Control (5 blocks +Mult.)	0.40	0.72	0.00112
MultiQAM (5 blocks +Mult.)	1.00	0.70	0.00197
TOTAL	1.40	1.41	0.00281

TABLE 5.5 – Power Results for Full OFDM chain

	Total on chip (mW)	Dynamic Power (mW)
Circuit Design with No DPR	0.246	0.04448
Circuit Design with DPR	0.242	0.03615

### 5.3 Real-Time Measurements and Analysis of Dynamic and PR Power Consumption for Zynq SoC Devices

It is interesting in our work to calculate the overhead power consumption for the PR operation on PS and PL. In order to evaluate the impact of dynamic and partial reconfiguration, we propose to design a prototype and measure the overhead power consumption when dealing with partial reconfiguration.

In this section, we present a SW/HW system implemented on a ZYNQ device. Dynamic and partial reconfiguration can be applied on a part of the circuit running on the PL. The advantage of a such platform is to measure the power consumed both by the FPGA and the processor during the partial reconfiguration process in order to obtain quantitative results.

#### 5.3.1 BackGround and Related Work

The DPR technique, which provides the ability to reconfigure a part of the system while the remaining parts are still running, may be seen as an approach to reduce the energy budget. This is achieved by reducing the amount of resources by dynamically time-multiplexing functions of the design.

Many studies have been performed for measuring and estimating the power consumed by FPGAs in general and when applying the partial reconfiguration process in particular. In [BHU03], the authors have measured and estimated the power consumption for a Virtex II Xilinx FPGA. A resistor shunt circuit is used to measure the consumed current and deduce the power consumption. The au-

thors have presented the results for the  $V_{cc}$  (board) and  $V_{core}$  (FPGA only) in both cases, full and partial reconfiguration. The reconfiguration procedure is done through the Xilinx Software from the computer.

The authors in [BCBS12] have proposed different mathematical models to estimate the partial reconfiguration power consumption in a Virtex-5 Xilinx FPGA. The PR process is performed by a MicroBlaze processor implemented on the FPGA through the ICAP interface. Power measurements are performed using a resistor shunt, a high precision amplifier and a digital oscilloscope.

In [BNY14], a new technology is used. The authors benefit from the Power Management Bus (PMBus) controllers embedded in the recent boards such as ZYNQ ZC702 and ZYNQ ZC706 of Xilinx's 7 series FPGAs. The authors have proposed two possible ways to access this power controller. The first consists in using the embedded hard processor (PS). The second makes use of the MicroBlaze soft processor and an I2C controller implemented on the PL. The power consumed by the PS and PL are presented according to different voltage levels. However, authors in [BNY14] did not present any measurements when running a partial reconfiguration process.

In our work, we have used a different methodology to measure the power consumption. Different partial reconfiguration techniques (PCAP) and recent ARM based SoC-FPGA devices from Xilinx (ZynQ) are considered. A program running on the embedded ARM processor manages the partial reconfiguration on FPGA. This is done by transferring the partial bit-streams from the external memory to the FPGA through the PCAP interface.

### 5.3.2 Experimental Setup

The ZC706 ZYNQ and the ZC702 ZYNQ evaluation boards have been used in our work. The ZC706 board has a ZYNQ 7000-XC7Z045 AP-SoC device, whereas the ZC702 board has a ZYNQ 7000-XC7Z020 CLG484 AP SoC one. The ZYNQ 7000-XC7Z045 device provides more resources than the ZYNQ 7000-XC7Z020. In both ZYNQ devices, PS and PL are integrated with high bandwidth AMBA AXI interconnection interfaces.

The DevCfg module, integrated on the PS, allows transferring partial bit-streams to the PL. The DevCfg configures the hardware through the 32-bit PCAP interface which is clocked at 100MHz [Koh13]. Another important feature of the two boards is the PMBus controller which provides a real-time monitoring of the voltage and current values from the regulators that are available on the board.

On the ZC706 board, the UCD90120A power monitor and controller from Texas Instruments (TI) enable to monitor the voltage and current of the FPGA PL only. However, the ZC702 board contains the UCD9248 Texas Instruments power controller that permits to monitor the voltage and current on both PS and PL simultaneously. The two power controllers are accessed through the I2C interface

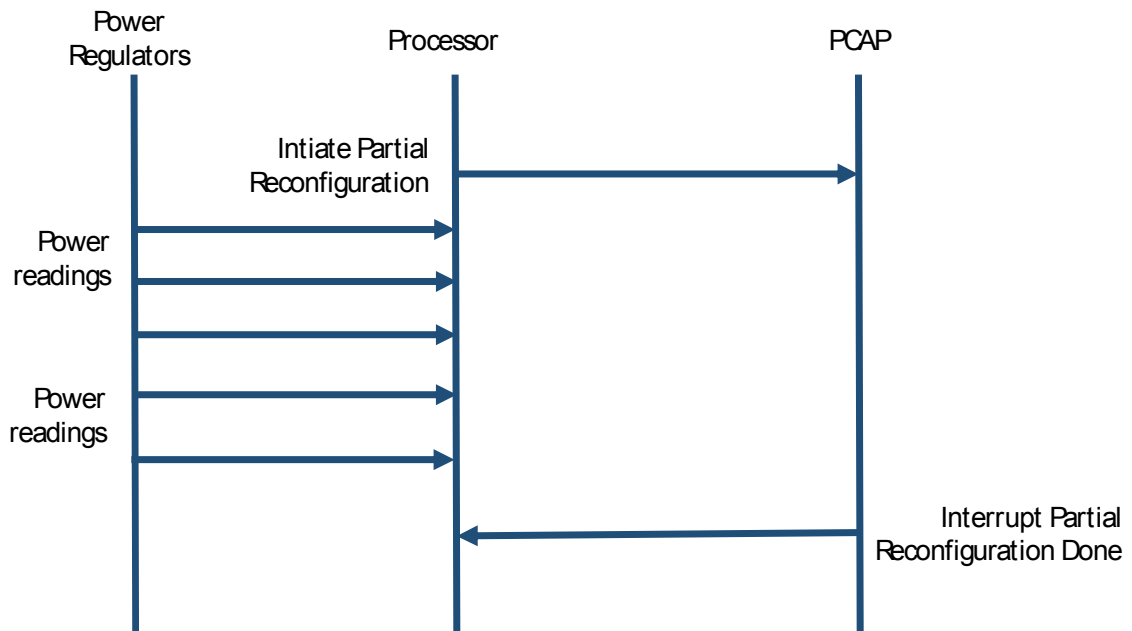


FIGURE 5.4 – Reading power values during Partial Reconfiguration

using the PMBus protocol [pmb]. There are two possible ways to obtain values from the PMBus controller. First, TI provides the Fusion Digital Power Designer software package with the GPIO/USB adapter to monitor real-time power values [TiF]. The second method consists in running a software application on the processor to access the power controller and read the voltage and current from multiple power rails on board. In our work, we adopted the two methods. We took benefit from the TI tool to get an overview of the power consumption on PL and PS during partial reconfiguration. Also, we run our implemented code on the processor to control the data to be sent, and to read these data only during the partial reconfiguration process.

In order to get power measurements samples during the partial reconfiguration phase only, the DevCfg driver was modified. Following the beginning of the partial bit-stream transfer, the power controller is accessed by the I2C bus and the process of measuring and extracting the current samples is started. This process continues

until an interrupt from the DevCfg is applied. This interrupt occurs when the reconfiguration of the concerned block of the FPGA is done. This process provides a precise estimate of the power consumption during the partial reconfiguration process. FIGURE 5.4 describes how power samples are collected during the PR process. The same operation is repeated many times in order to get a large number of measures samples. In Listing 5.1, it is shown how current values are retrieved during the partial reconfiguration process and before the operation is done.

Listing 5.1 – Reading current values during partial reconfiguration process

```

1 while ((IntrStsReg & XDCFG_IXR_D_P_DONE_MASK) != XDCFG_IXR_D_P_DONE_MASK)
2     {IntrStsReg = Xdcfg_IntrGetStatus(Instance);
3       current = readCurrent_real(device, page) ;
4     }
5
6 return XST_SUCCESS ;

```

The voltage and current measurements from the power regulators on the board are explained in this paragraph. The power controller chip communicates with the ZYNQ chip through an I2C electrical interface using the PMBus command protocol. On the two boards, the power controllers were on the same I2C channel 7. The PMBus is an open standard power management protocol, thus the PMBus commands were the same for monitoring current and voltage. The I2C addresses of the power controllers of the chip on the ZC702 board are 52 to 54. While the address is 101 on the ZC706 board. Unlike ZC702 board that can monitor the power consumed by the processor and the FPGA, the ZC706 board can only monitor the power consumed by the FPGA. The voltage and current are monitored every 1 second by default. But this can be modified.

The voltage is calculated according to the following equation :

$$Voltage = V * 2^N$$

Where V is the voltage value and N presents the exponent value. Read the V value by sending the READ\_VOUT command (0x8B) on the PMBus, a reply of two bytes is received and the V values are constructed from the value of the two bytes. Reading exponent value N is performed by sending the VOUT\_MODE command 0x20.

The current is calculated according to the following equation :

$$Current = I * 2^X$$



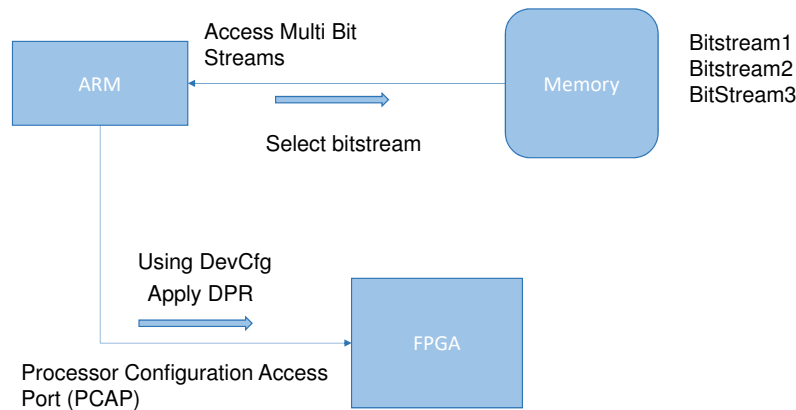


FIGURE 5.5 – Partial Reconfiguration from PS using PCAP

Where  $I$  presents the current values and  $X$  is the exponent value. Read the  $I$  value and the exponent  $X$  by sending the `READ_IOUT (0x8C)` and we get a reply of a two-byte command that contains the exponent  $X$  and the  $I$  value. Now we have the right voltage and current values for each iteration, the power consumed by the FPGA can be computed.

As an example of application, a simple re-configurable OFDM transmitter chain has been implemented on the PL. The main OFDM transmitter chain blocks are : the channel encoder, the modulation, the mapper, and the IFFT. This application has been taken from a part of a previous study and corresponds to a real case in communications systems [CCEPO12].

The overall system was implemented and the bit-streams were generated using the Vivado Xilinx software. On the ZC706, four OFDM transmitters were implemented as static parts, and one as a re-configurable module. Then, this re-configurable OFDM block was partially reconfigured while the other four identical OFDM transmitters were running in real-time. For the ZC702, we considered the whole OFDM chain as a re-configurable module with two different configurations. The sizes of the partial bit-streams used on the ZC706 and ZC702 boards are 807KB, and 724KB respectively.

Using Xilinx SDK, we have implemented a C application that runs on the PS part. This application allows to access the DevCfg driver in order to transfer partial bit-streams from memory to PL. At the software initial state, the partial bit-streams are transferred from the SD card to the RAM memory (DDR). Then, the partial bit-streams will be directly accessed from the memory during run-time as shown in FIGURE 5.5.

### 5.3.3 Experimental Results

The PMBus power controllers monitor and control multiple power regulators on the board. In our study, we will only focus on the power regulators described in TABLE 5.6 which will give an overview of the power consumed on the PS and the PL parts. The voltage values are constant approximately, for that the current value is enough to calculate the consumed power. The measured values are read from the power regulators with a sampling period of about 1 ms which corresponds to the minimum period that we have obtained.

Regulator	Voltage	Description
VCCINT	1.0V	powers all PL internal logic circuits
VCCPINT	1.0V	powers all PS internal logic circuits
VCCPAUX	1.8V	powers all PS auxiliary circuits

TABLE 5.6 – Power Regulators on board

The power consumed by the FPGA was measured for both ZYNQ devices that have different sizes. This was performed in order to evaluate the impact of the logic fabric size on the power consumption during partial reconfiguration.

#### 5.3.3.1 PL Power Measurements

First, the power consumed by the FPGA on the ZC706 board was measured. As previously explained, only the power consumed by PL part can be evaluated. The measurements were performed by a software application running on an ARM processor. The measured power values were sent (via the UART interface) to a Matlab application running on a PC. To evaluate the impact of PR on the power consumed by the FPGA, the power samples were collected only during the partial reconfiguration process by a modified version of the DevCfg driver. These power samples were compared to those obtained when the PR process was not activated. The results are provided in FIGURE 5.6 and FIGURE 5.7. Note that we considered the same static parts for the two scenarios to observe any additional power consumption when applying partial reconfiguration.

It can be clearly noticed that the values of the obtained samples in both configurations (DPR and no DPR) are almost the same. Since we have the same static part in both scenarios, we have approximately the same average power consumption. The average power of samples in FIGURE 5.6 is 274,8 mW, and the average

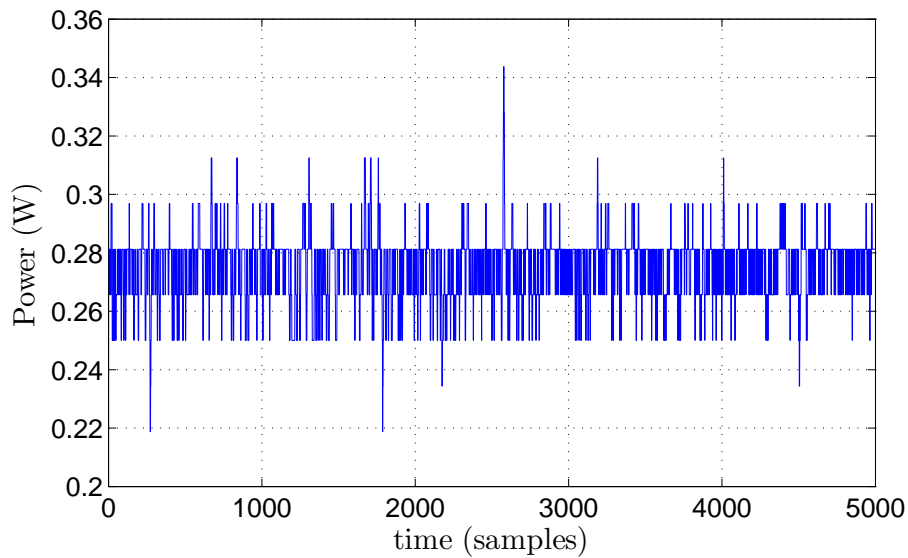


FIGURE 5.6 – Power samples only during the DPR process for the ZC706 Board

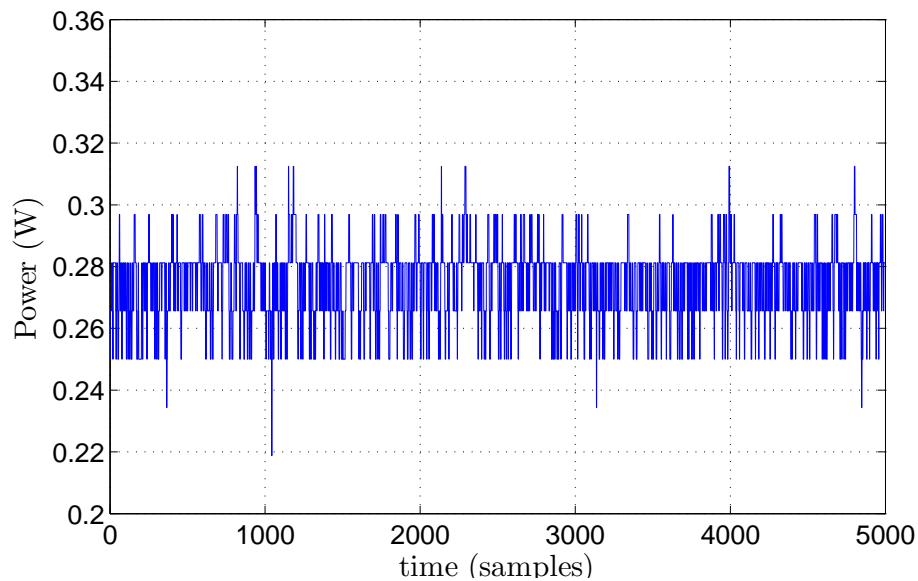


FIGURE 5.7 – Power consumed by PL while no DPR applied for the ZC706 Board

power of those in FIGURE 5.7 is 274,4 mW. The experiment was repeated multiple times giving approximately the same results every time and providing no clue about which process consumes the most power.

Next, we measure the power for PL on the ZC702 board. The power results

graphs are provided by the TI GUI tool. Similarly to the ZC706 scenario, the power consumed by the FPGA is measured during the two processes (with and without DPR). The results obtained for the power consumed by the VCCINT power regulator are similar to those obtained on the ZC706 board.

For both processes, the graphs representing the power consumption as a function of time are quite similar for the two boards. Using ZC706 board we have obtained an average value of 95 mW with DPR and 94.6 mW without DPR.

### 5.3.3.2 PS Power Measurements

For the PS part, the DPR process is compared to a reference process where the partial reconfiguration is not applied. Therefore, the DevCfg driver with the PCAP interface is not used. This reference process consists in running an infinite and empty while loop.

The DPR process is a C code running on the processor and has access to the DevCfg driver to apply consecutive partial bit-stream transfers to the PL. In this case, the PCAP interface is engaged in the PR process and the ARM processor is sending repetitive requests to perform the partial reconfiguration on the concerned block. The power consumed by the VCCPINT is measured using the TI GUI tool. FIGURE 5.8 and FIGURE 5.9 show the power measurements while running both the DPR process and the reference process, respectively.

As shown in FIGURE 5.8 and FIGURE 5.9, the power consumed during the reference process is slightly higher than that obtained for the DPR process. This result shows that there is no increase in power consumption when executing the partial reconfiguration process on the processor. The power is even lower due to the interrupts sources required for the PCAP management that blocks the code from execution until the completion of the PR operation. On the other hand, continuous memory accesses occur in the reference process.

In addition, we measured the power consumed by the VCCPAUX regulator during the DPR and the reference process. The auxiliary circuits consists of configuration section, input amplifiers, and input buffer circuits of some IO standars. FIGURE 5.10 shows that there is a small extra power consumed during both the DPR process. While running power measurements using the TI GUI tool, the process running on the PS is switched from the reference process to the DPR process. Power measurement samples were collected during 60 seconds. During the DPR process, 10500 partial bit-streams (708 KB each) transfers had been applied to consider the critical case of DPR. This is shown in FIGURE 5.10, where higher po-

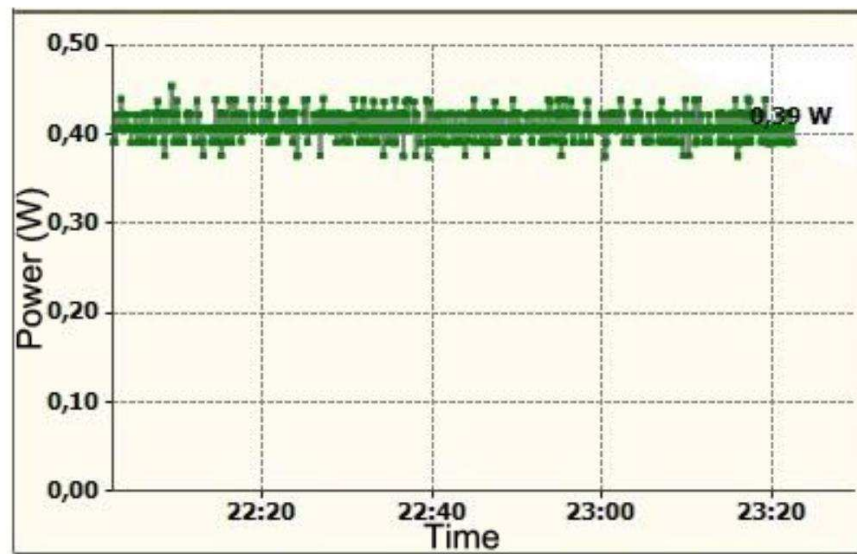


FIGURE 5.8 – Power consumed by PS running the DPR process

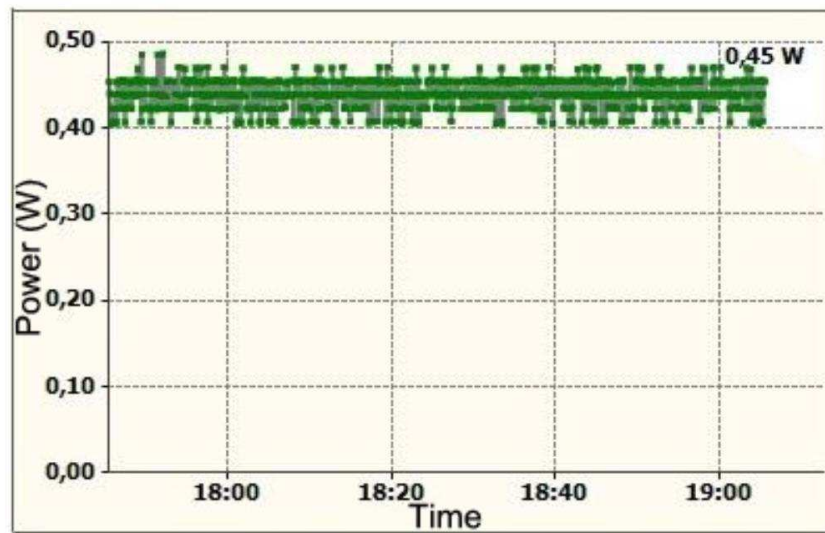


FIGURE 5.9 – Power consumed by PS internal circuits while running the reference process

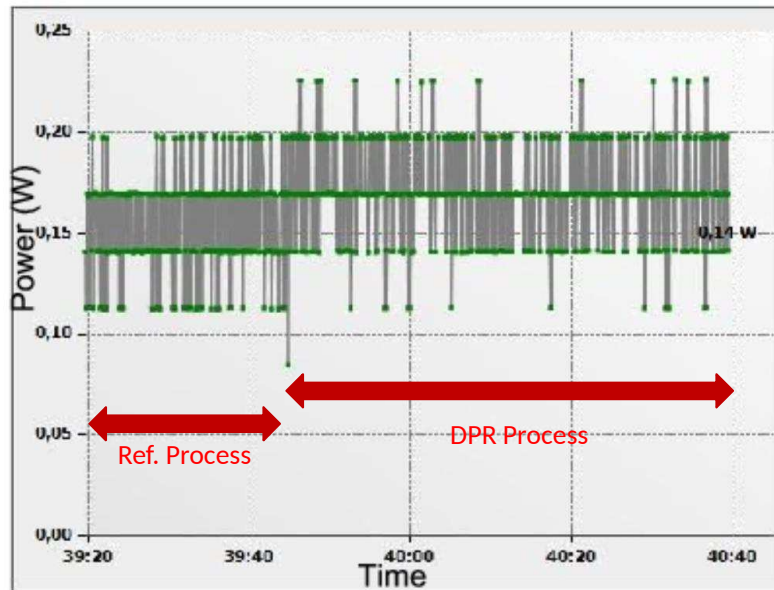


FIGURE 5.10 – Power consumed by VCCPAUX during the execution of both processes

wer values appear when launching the DPR process repetitively. The experiment was repeated many times providing similar results. The average power for the reference process is 157.7 mW whereas it is 170.3 mW when performing the DPR process. This additional power consumption is considered as the overhead power when applying partial reconfiguration through the PCAP interface. Despite this increase, the overhead remains relatively low (less than 20 mW). The average values for the power samples obtained in this section are summarized in TABLE 5.7.

Power Reg	with DPR	No DPR
VCCINT	274,8 mW	274,4 mW
VCCPINT	40.2 mW	41.8 mW
VCCPAUX	170.3 mW	157.7 mW

TABLE 5.7 – Average Power for PS and PL in two Cases

## 5.4 Summary

In this chapter, first we have explained in details how the partial reconfiguration technique is applied on ARM-FPGA platform in context of reconfigurable radio systems. We worked on adaptive modulator within OFDM SISO chain and compared the PR design with the parallel one. The two designs were compared in terms of on chip power consumption.

Then, we have described an experiment aiming at evaluating the power consumption of the dynamic and partial reconfiguration process in recent FPGA devices. The power measurements were done for the processor and the hardware logic fabrics in real time. It is noticed that the power consumed by the FPGA is almost unaffected when applying the partial reconfiguration. This result was obtained using two different Xilinx ZYNQ-7000 SoC devices. Also, the PS internal circuit does not exhibit any valuable additional power consumption. However, a slight increase in power consumption was noticed in the PS auxiliary circuits when the DPR is engaged. It was important to make sure that the power overhead is low and acceptable for miniaturized embedded systems.

# CHAPTER 6

## Conclusion and Perspectives

### 6.1 Summary

In the last decade, wireless communication systems have greatly evolved in terms of mobility, covered range, and throughput. A lot of standards with different specifications, both in the PHY (bandwidth, throughput, and modulation encoding techniques) and MAC layers, have been provided to meet different technical requirements and give access to new services.

The wireless channel is one of the most challenging communication channels. To optimize the usage of the available channel capacity, wireless communication chain should be reconfigured to adapt to the wireless channel characteristics. Also today, users are facing a huge diversity of communication networks and protocols that are generally very complex and heterogeneous. This is especially true for mobile devices that have to deal with Multi-RAT, during their mobility.

Hardware reconfiguration of wireless communication systems is the most interesting solution for implementing adaptive techniques and VHA. In this case, some system functions may be modified during runtime, while other functions continue to run without any interruption. FPGAs constitute the ideal circuits to implement such reconfigurable systems. Such devices are now sufficiently mature to implement very complex systems with a high level of performance. PR is one of the interesting features that has been added by FPGAs' vendors to ensure even more flexibility.

Another interesting feature in recent FPGA devices is the presence of embedded hard processor cores that are implemented with the hardware FPGA fabric, in the same chip. For example, the ZYNQ SoC devices from Xilinx feature a dual-



core ARM Cortex-A9 based (PS) as well as a PL within a single device. Compared to their soft core processors counterparts, these processors offer much more computing power and speed. SoC devices are appropriate for designing joint software and hardware systems. The existence of such systems in the end-devices allows designing high performance smart communication objects.

In this thesis, we present an ARM-FPGA based platform for reconfigurable wireless communication systems that benefits from the partial reconfiguration technique. The proposed system was tested for auto adaptive OFDM chain. It was also applied for switching between WiFi and WiMax networks based on machine learning VHA. Real time power measurements and demonstrations was performed to retrieve experimental results and test the proposed system.

In Chapter 1 have introduced the main concepts and the related work in the context of applying partial reconfiguration technique on ARM-FPGA platform for reconfigurable wireless systems.

In Chapter 2, an ARM-FPGA platform has been proposed for an adaptive wireless communication system. The system implemented on the the FPGA is partially reconfigured. This HW/SW platform is based on a custom micro-kernel that is already available in our laboratory. Its role consists in managing and controlling partial reconfiguration processes. Multiple software processes based on theoretical adaptive algorithms run in parallel and perform partial reconfiguration operations. These processes get knowledge of the environment condition by requesting parameters from the Parameter Provider. In this work, we made extensive use of the PCAP interface and optimized the size of the partial bit-streams in order to minimize the time overhead caused by partial reconfiguration. Also, to avoid data loss, the Configuration controller controls the data flow on the FPGA when initiating the PR process. An adaptive OFDM transmitter is implemented to test the proposed system. The obtained results have shown that implementing the PR technique with the micro-Kernel has decreased the global on-chip power consumption and reduced the reserved resources on the FPGA.

In Chapter 3, an ARM-FPGA platform proposed for auto adaptive wireless communication system is applied now to switch between multiple wireless standards. The general system design blocks on PS and PL where presented and explained in the current case. The mechanism of applying the partial reconfiguration technique in a unified reconfigurable chain for multiple wireless communication systems is discussed. Two reconfigurable architectures (ORBA and MRBA) are proposed and compared. The advantages and disadvantages of these architectures are presented. Selecting the best reconfigurable architecture depends on the desired

application. Where MRBA is considered more appropriate for wireless standards with similar PHY layer architecture. On the other hand ORBA is better for wireless standards with different PHY layer architecture. To test the proposed ARM-FPGA system and evaluate the two proposed architectures, a unified chain for WiFi and WiMax is considered as a case study. The similarities and differences of the two standards is studied to obtain a reconfigurable unified chain for both standards. The experimental results for the case study is retrieved by implementing and testing the unified chain on the ZedBoard. It was clearly shown the advantages of using PR technique to reconfigure unified chain for multiple wireless standards.

In Chapter 4, we have proposed a VHA for wireless standards based on adaptive scoring system with initial fast decision. Then we have proposed an NN based VHA to switch between multiple wireless standards in a heterogeneous network. The NN is combined with reinforcement learning to achieve better performance. A learning agent has been proposed to train the NN system with best records only to improve the NN decision. These best records were tagged based on the reward received after each decision made. Experimental results have been presented based on a WiFi-WiMax scenario. The results showed that the learning methodology with the use of reinforcement technique provides better performance in the NN system. The system was integrated within the previously proposed ARM-FPGA system which uses PR technique to switch between multiple wireless systems.

In Chapter 5, first we have explained more deeply how the partial reconfiguration technique is applied on ARM-FPGA platform in context of reconfigurable radio systems. We have worked on adaptive modulator on OFDM SISO chain and compare the PR design with the parallel design. The two designs are compared in terms of power on chip consumption. Then, we have described an experiment aiming at evaluating the power consumption of the dynamic and partial reconfiguration process in recent FPGA devices. The power measurements were done for the processor and the hardware logic fabrics in real time. It is noticed that the power consumed by the FPGA is almost unaffected when applying the partial reconfiguration.

Finally we presented two demos, one concerning the functionality of the Ker-One and the management of the PR processes. The important features of the micro-kernel is shown in an GUI interface which communicates with the ZedBoard to show real time data. The second demo illustrates the switching between WiFi and WiMax on ARM-FPGA platform using PR technique. The demo provides a GUI interface to simulate the mobility of an end-node and to show how the switching process is performed in real-time.

## 6.2 Perspectives and Future Works

Using dynamic partial reconfiguration technique on ARM-FPGA platform in context of reconfigurable wireless systems is considered as an interesting topic. In our study we have proposed multiple PS modules to control, update and manage reconfigurable blocks on the PL. We had also proposed creating unified reconfigurable chain for multiple wireless systems and the concept to switch PHY using PR technique. In context of the decision agent, we have proposed an intelligent VHA design based on Neural Networks and Reinforcement Learning. In our study we provided a real time measurements to show the small power overhead of the PR operation on PS and PL.

For the future work, it is interesting to have a real test for a wireless communication between two or multiple SoC based boards with applying PR to reconfigure the blocks in the wireless chain. It can be initially by reconfiguring simple blocks like the modulator to reach a full reconfigurable wireless chain.

It will be interesting also to apply a real testing for a switching between two wireless standards on two ARM-FPGA boards using partial reconfiguration technique. The switching between the wireless standards can be tested in real mobility scenario and be based on intelligent machine learning VHA.

Also it will be interesting to propose a combined architecture between PR and parallel design to optimize the best power and reconfiguration time for reconfigurable wireless systems. In the same context it can be proposed a unified reconfigurable chain for multiple wireless standards using hybrid design (PR and parallel design).

It will be interesting also to design a platform that compare multiple wireless standards and provide the used by the best reconfigurable unified architecture. This will be done by applying an algorithm that detect the similar and different blocks between the selected wireless standards and then propose the RMs in the PR design.

# Appendix A

## DEMONSTRATIONS

### A.1 Introduction

In this section we will present two use-cases. The first demonstration explains the functionality of the Micro-kernel used in the system. The second demonstration describes applying vertical handover between WiFi and WiMax using partial reconfiguration.

#### A.1.1 Ker-ONE : Embedded Virtualization Approach with Dynamic Reconfigurable Accelerators Management

Ker-ONE is a lightweight micro-kernel that provides paravirtualization on small-scaled embedded ARM systems. It cohosts multiple guest OSs in isolated virtual machines at the user level. These virtual machines are managed by a virtual machine monitor (VMM). Ker-ONE provides only fundamental functions such as scheduling, inter-process communication (IPC) and memory management, and ends up with a small trust computing base (TCB). The features of Ker-ONE include :

- Timing and spatial Isolation. Each VM domain is isolated from the others. Virtual machines run with virtualized resources provided by the VMM.
- Preemptive priority-based round-robin scheduler. In our case, the dedicated DPR resource manager service is given a higher priority than guest OSs, and preempts other running components as soon as it is scheduled.
- IRQ-based simple IPC Channel, that allows VMs to send messages to other components.

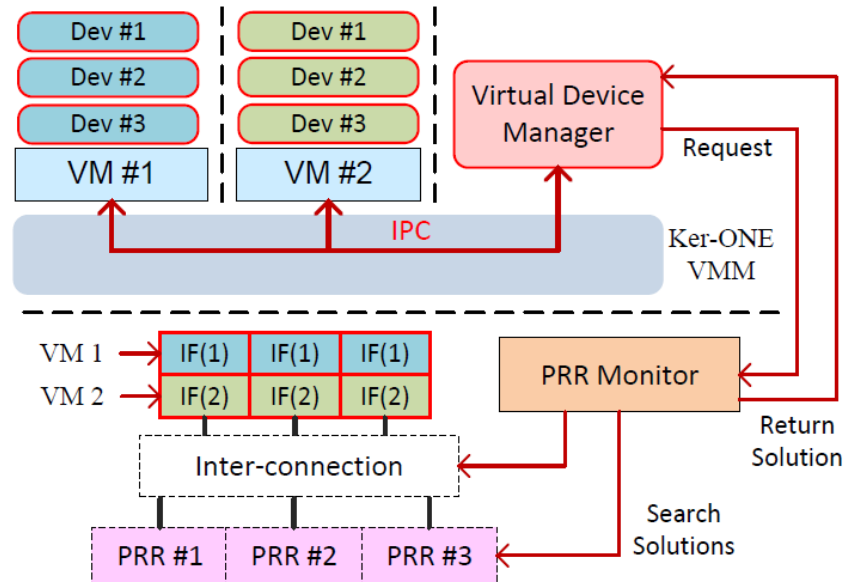


FIGURE A.1 – A plan that describes how Ker-One control and manages PR processes

Based on Ker-ONE, we have introduced a framework which facilitates the DPR resource management for virtual machines. The principle is to provide VM users with a highly-abstracted, transparent program interface to access reconfigurable accelerators. In our system, DPR accelerators are implemented in different partial reconfiguration regions (PRR) on the FPGA fabric. Between software and PRRs, we introduced an intermediate layer composed of PR interfaces (IF). In each virtual machine, IFs are mapped as universally-addressed accelerators, which can be accessed as ordinary peripherals. Each IF is exclusively associated to a specific virtual accelerator in a specific VM. Meanwhile, in lower layers, these IFs are connected to the DPR accelerators that actually perform the desired algorithm. Through dedicated memory management, our framework automatically detects the request for DPR resources and allocates them dynamically according to a preemptive allocation policy. Dedicated management components, i.e. Virtual Device Manager and PRR Monitor are implemented on both software and hardware sides to handle

allocations at run-time. The proposed management mechanism is depicted in FIGURE A.1. Note that the Virtual Device Manager is a particular software service in an independent virtual machine domain, which is given a highest priority level. Every time a VM requires a DPR accelerator, VMM immediately schedules the Virtual Device Manager to handle this DPR resource request.

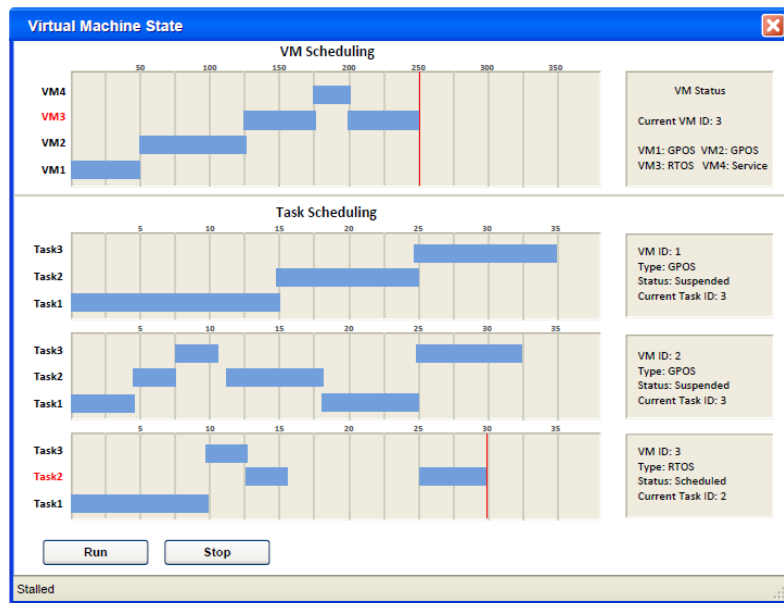


FIGURE A.2 – The concept of GUI presenting the status of virtual machines, where the scheduling of virtual machines, and guest OS tasks, are demonstrated with specific Gantt charts and time labels.

#### A.1.1.1 Demonstrations Description

The demonstration is built on the Xilinx Zynq-7000 SoC platform previously described. In the FPGA, several communication algorithms are implemented as reconfigurable accelerators, e.g. FFT and QAM. On top of Ker-ONE, multiple uC/OS-II execute concurrently in virtual machines as guest OSs with different priorities. In each guest OS, several tasks run periodically to use reconfigurable

accelerators. The demonstration demonstrates the dynamic allocation of DPR accelerators among different virtual machines. A custom GUI is implemented on the PC side, which is connected to the platform via the UART port, and is in charge of demonstrating the states of virtual machine and DPR resources. This GUI communicates with the Ker-ONE micro-kernel at run time and illustrates the procedures of software execution, and the allocation of accelerators. With this GUI, users are able to observe the scheduling of virtual machines and tasks in a step-in mode. Moreover, the usage of DPR resources, including their position, status and allocation, are dynamically and specifically presented in the GUI, so that the management algorithm can be clearly observed. The concepts of GUI are depicted in FIGURE A.2 and FIGURE A.3. During the demonstration the overhead of Ker-ONE functions is also calculated. It will be shown that the associated latency of VMM and DPR management is quite low (estimated as several micro-seconds).

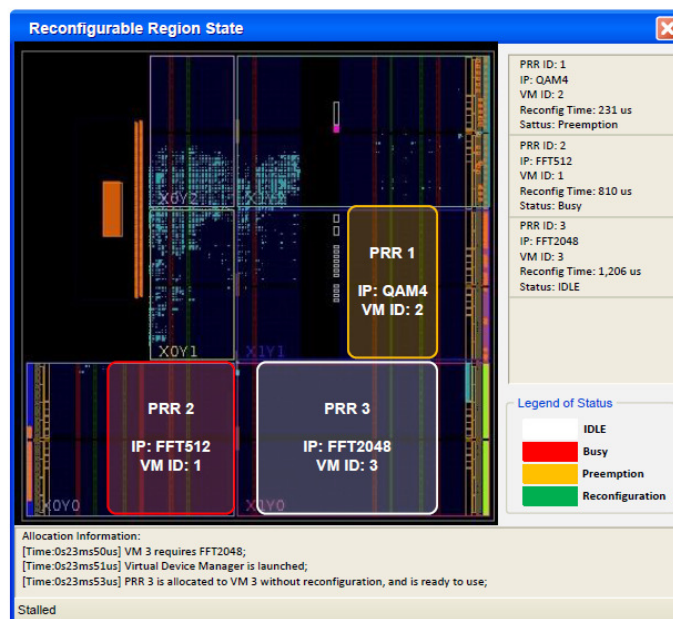


FIGURE A.3 – The concept of GUI presenting the status of reconfigurable accelerators, where the usage, position and status of PRRs are demonstrated. The GUI also prints the detailed procedure of allocation and the timing

### **A.1.2 WiFi-WiMax Vertical Handover on an ARM-FPGA Platform with Partial Reconfiguration**

In this demonstration we propose a system implementing previously proposed VHA for WiFi-WiMax wireless standards on an ARM-FPGA platform using the PR technique. The proposed system features a VHA running on the ARM. The unified reconfigurable chain that implements the PHY layers of the two wireless standards is implemented in hardware on the PL part of the FPGA. The demonstration simulates the mobility of the end-node, shows the decision made by the VHA and the status of the RMs on the FPGA. This demonstration is performed in real-time and is presented on an GUI interface.

#### **A.1.2.1 WiFi-WiMax Handover on ARM-FPGA Platform with PR Technique**

In our scenario, two IEEE wireless standards have been taken into consideration : WiFi and WiMax. These two standards have been chosen since they represent a real possible situation. A VHA running on the ARM is proposed to decide and select the best standard, then request reconfiguration of the unified reconfigurable chain implemented on the FPGA.

The VHA used in demonstration is described in CHAPTER 4. The physical layer for both standards have been studied to identify the similarities and differences between physical layers. The main architecture of the physical layer for both standards is based on the OFDM technique.

#### **A.1.2.2 Demonstration Description**

The demonstration can be divided into two parts. The first part consists of a C code application running on the ARM processor with a wireless communication chain implemented on the FPGA. The second part is a JAVA code running on the PC to implement a graphical interface. The two parts are connected through to exchange data in real-time. The demonstration is described in FIGURE A.4.

#### **A.1.2.3 VHA on ZedBoard with Partial Reconfiguration**

The first part of the demonstration is implemented on the ZedBoard that features a ZYNQ programmable SoC device from Xilinx. The proposed VHA runs on the PS with other modules for managing and controlling the PR process. Using Vivado tools, a reconfigurable unified chain for both WiFi and WiMax standards has



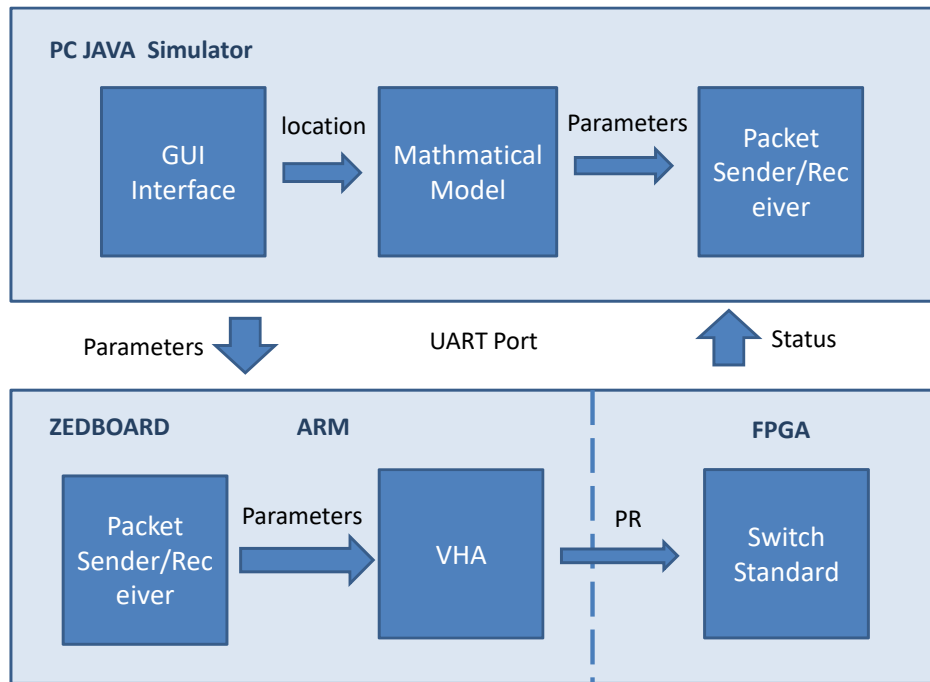


FIGURE A.4 – A plan that describes the demonstration and how the ZedBoard and the Simulator interacts

been implemented on the PL part. Multiple modules in the reconfigurable chain have been selected as RMs (Scrambler, Encoder, Interleaver, and IFFT). These modules are dynamically reconfigured when the VHA decides to switch from a standard to another based on MRBA design.

#### A.1.2.4 End-Node Mobility Simulator on the PC

The mobility of the end-node device in the WiFi-WiMax heterogeneous network is simulated using a JAVA FX application. The simulator shows the end-node device moving into different zones covered by one WiMax access point and multiple WiFi access points. Multiple parameters are collected in real-time and during the mobility of the node. Examples of such parameters are the speed and the RSS. Using the GUI interface, the user can also modify some parameters such as the battery level, user application, and node speed. Also, the interface displays the decision made by the VHA and the status of the RMs on the ZedBoard. In FIGURE A.5, we present the overall design of the demonstration with the reconfigurable unified

chain on the PL.

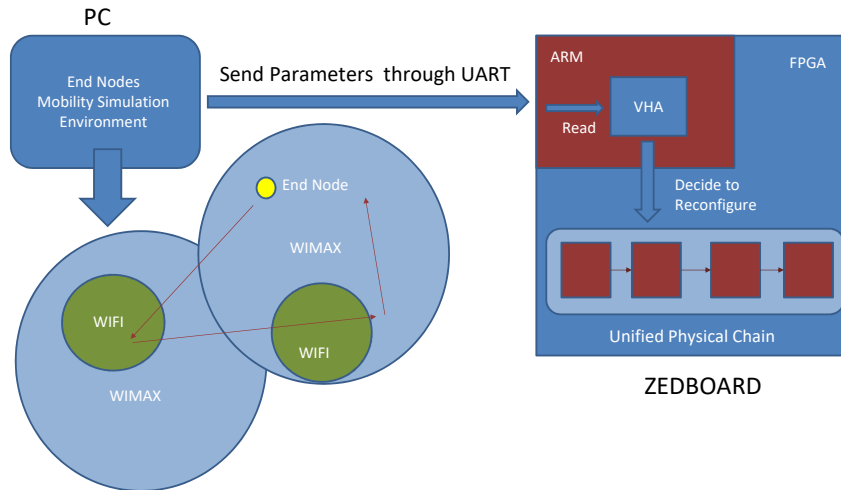


FIGURE A.5 – A plan that illustrates the overall functionality of the demonstration showing the reconfigurable blocks on PL

#### A.1.2.5 Connection between the ZedBoard and the Simulator

The simulator on the PC and the application running on the ZedBoard communicates through the UART. The simulator collects the parameters' values generated from mathematical models in a packet and transmit it to the application on the ARM. The application extracts the values from the packet and applies the VHA to select the best standard. If the VHA decides to switch from a standard to another, it will request the reconfiguration controller module to reconfigure the unified chain. On the ZedBoard, the application transmits the new status of the configurable modules to the simulator. This transmission is also illustrated in the GUI Interface (see FIGURE A.6).

On the Java GUI interface the user can select different scenarios. Each scenario affects the mobility of the node. For example, in a selected scenario the node is initially connected to WiMax and moving at a high speed in a WiFi access point coverage. The VHA is triggered when RSS of WiFi is greater than that of WiMax. Then VHA reads the speed parameter and considers this as a fake handover.

On the GUI interface a green point is drawn to represent the location where the fake handover was detected. In another scenario the node is located in the coverage of the WiFi access point. the user can change the power level and user preferences to notice how the VHA decides to switch from standard to another. When the communication is switched from standard to another, the color of the node is changed and a colored point is drawn in that location. Also the color of the hardware blocks of the unified chain illustrated on the FPGA figure ( as shown in FIGURE A.6) changes when the VHA makes a decision.

### FPGA Blocks:

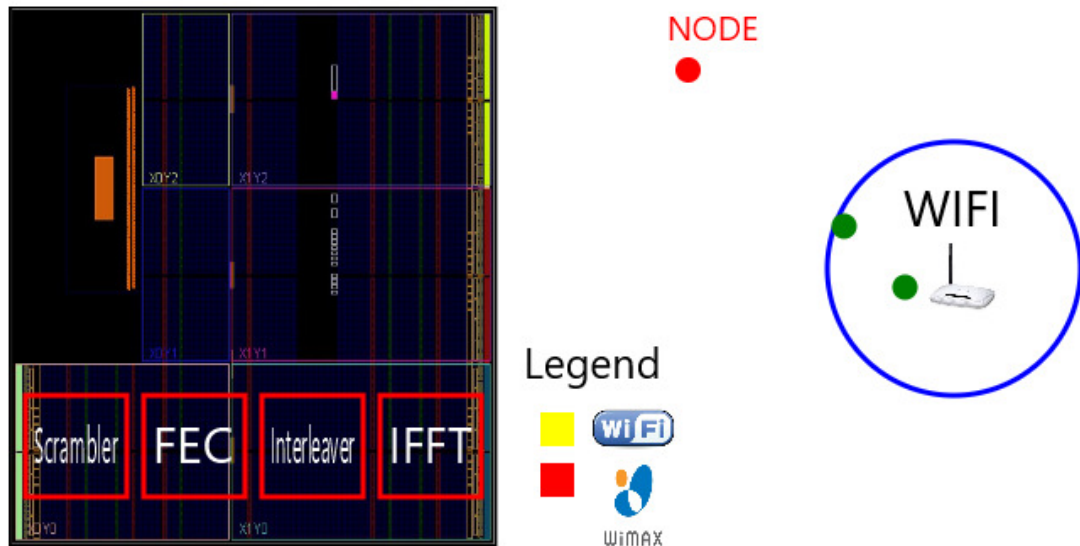


FIGURE A.6 – A screen-shot shows how the simulator illustrates the status of the reconfigurable blocks, the mobility end node and the VHA decision

# Appendix B

## EVALUATION BOARDS

### B.1 ZedBoard

ZedBoard (see FIGURE B.1) is a low-cost development board for the Xilinx Zynq-7000 All Programmable SoC. This board contains everything necessary to create a Linux, Android, Windows or other OS/RTOS-based design. Additionally, several expansion connectors expose the processing system and programmable logic I/Os for easy user access. Take advantage of the Zynq-7000 AP SoC's tightly coupled ARM processing system and 7 series programmable logic to create unique and powerful designs with ZedBoard. ZedBoard is supported by the [zedboard.org](http://zedboard.org) community website where users can collaborate with other engineers also working on Zynq designs.

#### B.1.1 FEATURES

- Zynq-7000 All Programmable SoC XC7Z020-CLG484-1
- Memory
  - 512 MB DDR3
  - 56 Mb Quad-SPI Flash
  - 4 GB SD card
- Onboard USB-JTAG Programming
- 10/100/1000 Ethernet
- USB OTG 2.0 and USB-UART
- PS and PL I/O expansion
- (FMC, Pmod™ Compatible, XADC)
- Multiple displays (1080p HDMI, 8-bit VGA, 128 x 32 OLED)

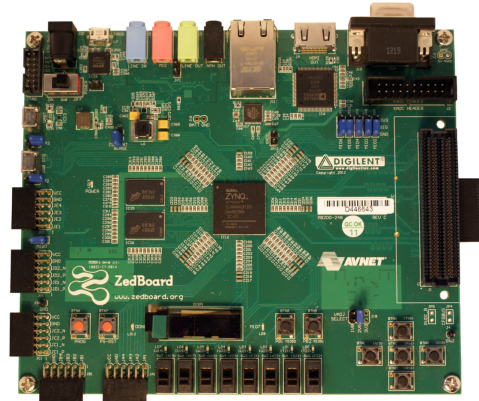


FIGURE B.1 – ZedBoard Evaluation Board

- I2S Audio CODEC

## B.1.2 TARGET APPLICATIONS

- Video processing
- Motor control
- Software acceleration
- Linux/Android/RTOS development
- Embedded ARM processing
- General Zynq
- 7000 All Programmable SoC prototyping

## B.2 ZYNQ-7000 SOC ZC706 Evaluation Kit

The Zynq™-7000 All Programmable SoC ZC706 Evaluation Kit (FIGURE B.2) includes all the basic components of hardware, design tools, IP , and pre-verified reference designs including a targeted design. This makes it a complete embedded processing platform and 12.5 Gbps transceiver for high-end applications that require higher performance and high I/O throughput. The included pre-verified reference designs and industry standard FPGA Mezzanine Connectors (FMC ) allow scaling and customization with daughter cards.

The Zynq-7000 AP SoC PCI Express Targeted Reference Design demonstrates PCI Express communication with a host system at PCI Express x4 GEN2 speed. In this PCI Express Targeted Reference Design, the input of the video processing

pipeline is generated by an application on the host computer at 1080p60 resolution and transmitted to the ZC706 board via PCI Express. The data is processed by the video pipeline implemented in Programmable Logic and passed back to the host system via PCI Express. As full 1080p60 video stream only take up around 4Gbps, an additional data generator and a checker are implemented and connected to channel 1 of the PCI Express DMA showcasing the maximum PCI Express x4 GEN2 bandwidth achieved by the hardware.

### B.2.1 Key Features and Benefits

- Optimized for quickly prototyping embedded applications using Zynq-7000 SoCs
- Hardware, design tools, IP, and pre-verified reference designs
- Demonstrates a embedded design, targeting video pipeline
- Advanced memory interface with 1GB DDR3 Component Memory and 1GB DDR3 SODIM Memory
- Enabling serial connectivity with PCIe Gen2x4, SFP+ and SMA Pairs, USB OTG, UART, IIC
- Supports embedded processing with Dual ARM Cortex-A9 core processors
- Develop networking applications with 10/100/1000 Mbps Ethernet (RG-MII)
- Implement Video display applications with HDMI out
- Expand I/O with the FPGA Mezzanine Card (FMC) interface

## B.3 ZYNQ-7000 SOC ZC702 Evaluation Kit

### Product Description

The Zynq-7000 SoC ZC702 Evaluation Kit (see FIGURE B.3) includes all the basic components of hardware, design tools, IP, and pre-verified reference designs including a targeted design, enabling a complete embedded processing platform. The included pre-verified reference designs and industry-standard FPGA Mezzanine Connectors (FMC) allow scaling and customization with daughter cards.

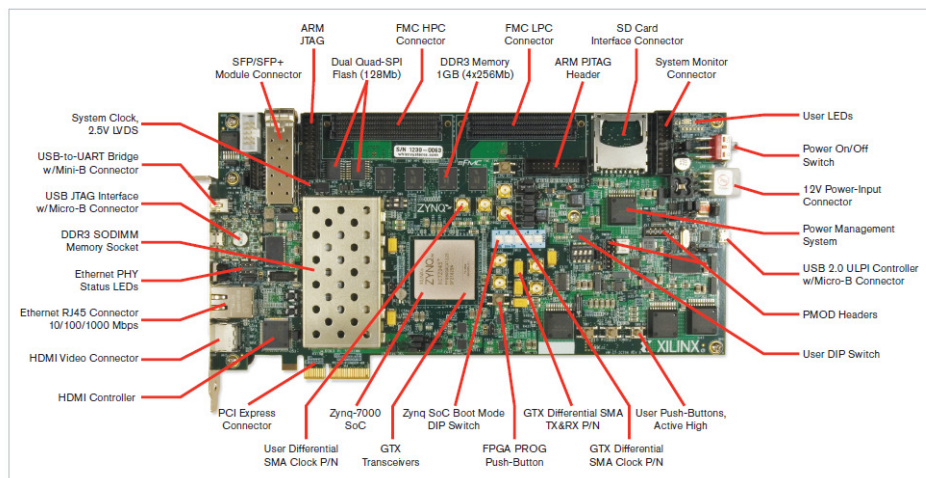


FIGURE B.2 – ZC706 Evaluation Board

### B.3.1 Key Features and Benefits

- Optimized for quickly prototyping embedded applications using Zynq-7000 SoCs
- Hardware, design tools, IP, and pre-verified reference designs
- Demonstrates an embedded design, targeting video pipeline
- Advanced memory interface with 1GB DDR3 Component Memory
- Enabling serial connectivity with USB OTG, UART, IIC, CAN Bus
- Supports embedded processing with Dual ARM Cortex-A9 core processors
- Develop networking applications with 10/100/1000 Mbps Ethernet (GMII, RGMII and SGMII)
- Implement Video display applications with HDMI out
- Expand I/O with the FPGA Mezzanine Card (FMC) interface



FIGURE B.3 – ZC702 Evaluation Board





# LIST OF FIGURES

1	Blocs de conception générale du système . . . . .	xi
2	Processus adaptatifs reconfigurant l'émetteur OFDM pendant l'exécution en fonction des valeurs des paramètres . . . . .	xiv
3	Liste des modules reconfigurables pendant VHA . . . . .	xvii
4	Etat des modules reconfigurables pendant le VHA . . . . .	xviii
5	Conception proposée de VHA . . . . .	xxi
6	Capture d'écran du simulateur illustrant l'état des blocs reconfigurables, le noeud mobile et la décision VHA . . . . .	xxv
1.1	Wireless channel effects on transmitted signals . . . . .	9
1.2	OFDM symbol as shown in frequency domain . . . . .	11
1.3	OFDM modulation and demodulation process . . . . .	11
1.4	Data rate vs. range for IoT connectivity technologies . . . . .	13
1.5	Reconfigurable radio block diagram . . . . .	14
1.6	Adaptive modulation performance results . . . . .	16
1.7	Vertical Handover in WiFi-WiMax Heterogeneous Network . . . . .	18
1.8	Neural network block diagram . . . . .	21
1.9	Reinforcement learning block diagram . . . . .	22
1.10	General proposed Vertical Handover (VHO) with R-MLP and fuzzy logic controller . . . . .	23
1.11	An example of switching the physical layer between two wireless systems . . . . .	24
1.12	Partial reconfiguration on FPGA example . . . . .	26
1.13	FPGA internal architecture . . . . .	26
1.14	Partial reconfiguration flow through PCAP interface . . . . .	28
1.15	Partial reconfiguration flow through ICAP interface and Microblaze . . . . .	29
1.16	A simplified architecture of the Zynq SoC . . . . .	29
1.17	ZedBoard with the Zynq-7000 SoC integrated with other peripherals . . . . .	31

2.1	General system design blocks . . . . .	38
2.2	Zynq block as represented in design flow using Vivado tools . . . . .	39
2.3	Dataflow controlled by the Configuration Controller during Partial Reconfiguration . . . . .	41
2.4	Parameter Provider collects parameters from different layers . . . . .	42
2.5	Hardware Updater updates versions of RMs . . . . .	43
2.6	OFDM transmitter base modules . . . . .	44
2.7	Floorplan view of the reconfigurable and static modules of the OFDM transmitter . . . . .	45
2.8	Adaptive OFDM transmitter implemented in the proposed design . . . . .	46
2.9	Parallel implementation for reconfigurable OFDM with multiple blocks . . . . .	46
2.10	Implementation for reconfigurable OFDM with multiple blocks using PR technique . . . . .	47
2.11	Flow chart for the adaptive modulation process . . . . .	48
2.12	Adaptive processes reconfiguring OFDM transmitter during runtime according to values of parameters . . . . .	54
3.1	General system design blocks in the scenario of unified chain for multiple wireless systems . . . . .	58
3.2	State of part of the RMs during VHA . . . . .	60
3.3	Partial reconfiguration of the the unified chain using ORBA . . . . .	62
3.4	State of the RMs during VHA . . . . .	65
3.5	State of the RMs during VHA . . . . .	70
4.1	From WiMax to WiFi Vertical Handover Algorithm . . . . .	76
4.2	From WiFi to WiMax Vertical Handover Algorithm . . . . .	77
4.3	Proposed VHA design . . . . .	81
4.4	Training and Validation System Response of the Neural Network System. . . . .	86
4.5	Rate of Successful Decisions using all records for Power Consumption. . . . .	88
4.6	Rate of Successful Decisions using tagged records only for Power Consumption. . . . .	89
4.7	Rate of Successful Decisions using all records (QoS) . . . . .	89
4.8	Rate of Successful Decisions using tagged records only (QoS) . . . . .	90
5.1	Parallel implementation for adaptive modulation in OFDM chain . . . . .	95
5.2	PR implementation for adaptive modulation in OFDM chain . . . . .	96

5.3	Boot sequence on the ZedBoard . . . . .	97
5.4	Reading power values during Partial Reconfiguration . . . . .	102
5.5	Partial Reconfiguration from PS using PCAP . . . . .	104
5.6	Power samples only during the DPR process for the ZC706 Board .	106
5.7	Power consumed by PL while no DPR applied for the ZC706 Board	106
5.8	Power consumed by PS running the DPR process . . . . .	108
5.9	Power consumed by PS internal circuits while running the reference process . . . . .	108
5.10	Power consumed by VCCPAUX during the execution of both processes	109
A.1	A plan that describes how Ker-One control and manages PR processes	116
A.2	The concept of GUI presenting the status of virtual machines, where the scheduling of virtual machines, and guest OS tasks, are demons- trated with specific Gantt charts and time labels. . . . .	117
A.3	The concept of GUI presenting the status of reconfigurable accele- rators, where the usage, position and status of PRRs are demons- trated. The GUI also prints the detailed procedure of allocation and the timing . . . . .	118
A.4	A plan that describes the demonstration and how the ZedBoard and the Simulator interacts . . . . .	120
A.5	A plan that illustrates the overall functionality of the demonstra- tion showing the reconfigurable blocks on PL . . . . .	121
A.6	A screen-shot shows how the simulator illustrates the status of the reconfigurable blocks, the mobility end node and the VHA decision	122
B.1	ZedBoard Evaluation Board . . . . .	124
B.2	ZC706 Evaluation Board . . . . .	126
B.3	ZC702 Evaluation Board . . . . .	127



# LIST OF TABLES

1	Taille partielle de Bitstream et temps de reconfiguration . . . . .	xiii
2	Comparaison de l'utilisation des ressources matérielles . . . . .	xiv
3	Comparaison de l'utilisation des ressources matérielles . . . . .	xvii
4	Puissance moyenne pour PS et PL dans deux cas . . . . .	xxiv
2.1	OFDM transmitter parameters . . . . .	44
2.2	Partial bit-streams size and reconfiguration time . . . . .	51
2.3	Size and power consumption of the considered blocks . . . . .	52
2.4	Comparison of hardware resource usage . . . . .	54
3.1	Comparison between both reconfigurable schemes . . . . .	61
3.2	Partial bit-streams size and reconfiguration time . . . . .	68
3.3	Size and power consumption of the considered blocks . . . . .	69
3.4	Comparison of hardware resources usage . . . . .	69
4.1	WiMax and WiFi networks comparison . . . . .	75
4.2	Example of a scoring system . . . . .	78
4.3	Scoring system example according to different applications . . . . .	80
4.4	Example showing how the learning agent selects the best records . . . . .	87
4.5	MSE of Training and Validation for different number of samples. . . . .	88
5.1	Power Results for modulator block only using DPR design . . . . .	99
5.2	Power Results for modulator block only using parallel design . . . . .	99
5.3	Power Results for QAM and Control blocks in full OFDM chain using DPR design . . . . .	99
5.4	Power Results for total control and QAM blocks using parallel design	100
5.5	Power Results for Full OFDM chain . . . . .	100
5.6	Power Regulators on board . . . . .	105
5.7	Average Power for PS and PL in two Cases . . . . .	109



# BIBLIOGRAPHY

- [AA15] N. M. Alotaibi and S. S. Alwakeel. A neural network based handover management strategy for heterogeneous networks. In *Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on*, pages 1210–1214. IEEE, 2015.
- [ABG14] A. Ahmed, L.M. Boulahia, and D. Gaiti. Enabling vertical handover decisions in heterogeneous wireless networks : A state-of-the-art and a classification. *IEEE Communications Surveys & Tutorials*, 16(2) :776–811, 2014.
- [All15] LoRa Alliance. Lorawan specification. *LoRa Alliance*, 2015.
- [alt] Altera corporation. <https://www.altera.com/>. Accessed : 2018-09-03.
- [AMM12] B. Ammar A., D. B. Mohd, and I. Muhammad. Ieee 802.21 based vertical handover in wifi and wimax networks. In *2012 IEEE Symposium on Computers Informatics (ISCI)*, pages 140–144, March 2012.
- [ARS10] A. Aziz, S. Rizvi, and N. M. Saad. Fuzzy logic based vertical handover algorithm between lte and wlan. In *2010 International Conference on Intelligent and Advanced Systems*, pages 1–4, June 2010.
- [BCBS12] Robin Bonamy, Daniel Chillet, Sébastien Bilavarn, and Olivier Sentieys. Power consumption model for partial and dynamic reconfiguration. In *Reconfigurable Computing and FPGAs (ReConFig), 2012 International Conference on*, pages 1–8. IEEE, 2012.
- [Bha07] PP Bhattachrya. Application of artificial neural network in cellular handoff management. In *Conference on Computational Intelligence and Multimedia Applications*, volume 1, pages 237–241. IEEE, 2007.
- [BHU03] Juergen Becker, Michael Huebner, and Michael Ullmann. Power estimation and power measurement of xilinx virtex fpgas : trade-offs and



- limitations. In *Integrated Circuits and Systems Design, 2003. SBCCI 2003. Proceedings. 16th Symposium on*, pages 283–288. IEEE, 2003.
- [BNY14] Arash Farhadi Beldachi and Jose L Nunez-Yanez. Accurate power control and monitoring in zynq boards. In *Field Programmable Logic and Applications (FPL), 2014 24th International Conference on*, pages 1–4. IEEE, 2014.
- [CCEPO12] Laura Conde-Canencia, Yvan Eustache, J-C Prévotet, and Yaset Oliva. Modeling adaptive coded modulation in real time partially reconfigurable mobile terminals. In *Signal Processing Conference (EU-SIPCO), 2012 Proceedings of the 20th European*, pages 1049–1053. IEEE, 2012.
- [CEES14] Louise H Crockett, Ross A Elliot, Martin A Enderwitz, and Robert W Stewart. *The Zynq Book : Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable Soc*. Strathclyde Academic Media, 2014.
- [CFY04] Yan Chen, Toni Farley, and Nong Ye. Qos requirements of network applications on the internet. *Information Knowledge Systems Management*, 4(1) :55–76, 2004.
- [Cha66] Robert W Chang. Synthesis of band-limited orthogonal signals for multi-channel data transmission. *Bell Labs Technical Journal*, 45(10) :1775–1796, 1966.
- [Chr16] M Bishop Christopher. *PATTERN RECOGNITION AND MACHINE LEARNING*. Springer-Verlag New York, 2016.
- [DF03] C. Desset and A. Fort. Selection of channel coding for low-power wireless systems. In *The 57th IEEE Semiannual Vehicular Technology Conference, 2003. VTC 2003-Spring.*, volume 3, pages 1920–1924 vol.3, April 2003.
- [DMN<sup>+</sup>08] J. Delorme, J. Martin, A. Nafkha, C. Moy, F. Clermidy, P. Leray, and J. Palicot. A fpga partial reconfiguration design approach for cognitive radio based on noc architecture. In *2008 Joint 6th International IEEE Northeast Workshop on Circuits and Systems and TAISA Conference*, pages 355–358, June 2008.
- [DP12] Brijesh Darji and Bhavna Pancholi. Implementation of unified architecture of 802.11 a and 802.16 a phy layers using verilog hdl (r & d). *organization*, 2(4) :573–576, 2012.

- [DPML07] J. P. Delahaye, J. Palicot, C. Moy, and P. Leray. Partial reconfiguration of fpgas for dynamical reconfiguration of a software radio platform. In *2007 16th IST Mobile and Wireless Communications Summit*, pages 1–5, July 2007.
- [FBF16] Mário Lopes Ferreira, Amin Barahimi, and João Canas Ferreira. *Reconfigurable FPGA-Based FFT Processor for Cognitive Radio Applications*, pages 223–232. Springer International Publishing, Cham, 2016.
- [FS] J. Faezah and K. Sabira. Adaptive modulation for ofdm systems. *International Journal of Communication Networks and Information Security (IJCNIS)*.
- [Gol05] Andrea Goldsmith. *Wireless communications*. Cambridge university press, 2005.
- [Han70] Per Brinch Hansen. The nucleus of a multiprogramming system. *Communications of the ACM*, 13(4) :238–241, 1970.
- [IEE99] IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications High-speed Physical Layer in the 5 GHz Band, 802.11a. <http://standards.ieee.org>, 1999.
- [JH09] I. Joe and S. Hong. An adaptive network scanning algorithm in hybrid wireless networks. In *Proceedings of the 4th International Conference on Ubiquitous Information Technologies Applications*, pages 1–4, Dec 2009.
- [JL15] Wang Junxuan and Kon Lingbin. Vertical handover method based on bayesian decision in heterogeneous networks. 2015.
- [KH00] T. Keller and L. Hanzo. Adaptive modulation techniques for duplex ofdm transmission. *IEEE Transactions on Vehicular Technology*, 49(5) :1893–1906, Sep 2000.
- [KJU<sup>+</sup>15] M. Khan, C. Jung, P. C. Uzoh, C. Zhenbo, J. Kim, Y. Yoon, A. Na-deem, and K. Han. Enabling vertical handover management based on decision making in heterogeneous wireless networks. In *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 952–957, Aug 2015.
- [KM09] Nate Kohl and Risto Miikkulainen. Evolving neural networks for strategic decision-making problems. *Neural Networks*, 22(3), 2009.

- [Koh13] Christian Kohn. Partial reconfiguration of a hardware accelerator on zynq-7000 all programmable soc devices. *Xilinx, XAPP1159 (v1. 0)*, 2013.
- [Kor11] Leonhard Korowajczuk. *LTE, WiMAX and WLAN network design, optimization and performance analysis*. John Wiley & Sons, 2011.
- [Kum12] K. A. Arun Kumar. Fpga implementation of psk modems using partial re-configuration for sdr and cr applications. In *2012 Annual IEEE India Conference (INDICON)*, pages 205–209, Dec 2012.
- [Kum13a] K. A. Arun Kumar. Fpga implementation of qam modems using pr for reconfigurable wireless radios. In *2013 Annual International Conference on Emerging Research Areas and 2013 International Conference on Microelectronics, Communications and Renewable Energy*, pages 1–6, June 2013.
- [Kum13b] K. A. Arun Kumar. Fpga implementation of qam modems using pr for reconfigurable wireless radios. In *2013 Annual International Conference on Emerging Research Areas and 2013 International Conference on Microelectronics, Communications and Renewable Energy*, pages 1–6, June 2013.
- [Kum14] K. A. A. Kumar. An ofdm transmitter implementation using cordic based partially reconfigurable ifft module. In *2014 3rd International Conference on Eco-friendly Computing and Communication Systems*, pages 266–270, Dec 2014.
- [lte] Lte. <http://3gpp.org>. Accessed 22 Feb 2018.
- [LTG04] Jian Liang, R. Tessier, and D. Goeckel. A dynamically-reconfigurable, power-efficient turbo decoder. In *12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, pages 91–100, April 2004.
- [LWL08] Tsungnan Lin, Chiapin Wnag, and Po-Chiang Lin. A neural-network-based context-aware handoff algorithm for multimedia computing. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 4(3), 2008.
- [LWN09] Q. Lu, X. Wang, and J. Niu. A low-power variable-length fft processor base on radix-24 algorithm. In *2009 Asia Pacific Conference on Postgraduate Research in Microelectronics Electronics (PrimeAsia)*, pages 129–132, Jan 2009.

- [MM99] Joseph Motila and Gerald Q Maguire. Cognitive radio : making software radios more personal. *IEEE personal communications*, 6(4) :13–18, 1999.
- [MN15] Marutham M and L. Nandita. Performance improvement of ofdm-based cognitive radio system. In *2015 International Conference on Communications and Signal Processing (ICCSP)*, pages 1623–1627, April 2015.
- [NADS16] B. Ng, Y. Qu A. Deng, and W. K. G. Seah. Changeover prediction model for improving handover support in campus area wlan. In *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*, pages 256–272. IEEE, 2016.
- [NGS07] Nidal Nasser and Eyhab Guizanim Sghaier, Al-Masri. Middleware vertical handoff manager : A neural network-based solution. In *IEEE International Conference on Communications, 2007, ICC'07.*, 2007.
- [NR15] A. K. Nahar and Y. Rahayu. Design and implementation of mc-cdma wireless communication system using partial reconfiguration in fpga—a review. In *Proc. of the Third Intl. Conf. on Advances in Computing, Electronics and Electrical Technology - CEET, USA*, 2015.
- [NVD<sup>+</sup>07] M. C. Ng, M. Vijayaraghavan, N. Dave, Arvind, G. Raghavan, and J. Hicks. From wifi to wimax : Techniques for high-level ip reuse across different ofdm protocols. In *2007 5th IEEE/ACM International Conference on Formal Methods and Models for Codesign (MEMOCODE 2007)*, pages 71–80, May 2007.
- [PA10] P. S. Prasad and P. Agrawal. Movement prediction in wireless networks using mobility traces. In *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, pages 1–5. IEEE, 2010.
- [pmb] Pmbus. <http://pmbus.org/index.php>. Accessed 23 March 2016.
- [RMGV14] Rapeepat Ratasuk, Nitin Mangalvedhe, Amitava Ghosh, and Benny Vejlgaard. Narrowband lte-m system for m2m communication. In *Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th*, 2014.
- [RRMS11] Devashish Raval, Nilesh Ranpura, Ekata Mehul, and Zuber Saiyed. Simulation of unified architecture of ieee 802.11 a and 802.16 a phy layers using matlab. *International Journal of Computer Science & Emerging Tech-nologies (E-ISSN : 2044-6004)*, 272, 2011.

- [SABN14] S. Sivanantham, R. Adarsh, S. Bhargav, and K. J. Naidu. Partial reconfigurable implementation of ieee802.11g ofdm. *Indian Journal of Science and Technology*, 2014.
- [SPA10] A. Sharif, V. M. Potdar, and R. F. Ahmad. Adaptive channel coding and modulation scheme selection for achieving high throughput in wireless networks. In *2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*, pages 200–207, April 2010.
- [TiF] Ti fusion. [http://focus.ti.com/docs/toolsw/folders/print/fusion\\_digital\\_power\\_designer.html](http://focus.ti.com/docs/toolsw/folders/print/fusion_digital_power_designer.html). Accessed 22 March 2016.
- [uco] ucos. <http://micrium.com/rtos>. Accessed 7 July 2018.
- [VSR<sup>+</sup>13] C. Vennila, K. Suresh, R. Rather, G. Lakshminarayanan, and Seok-Bum Ko. Dynamic partial reconfigurable adaptive transceiver for ofdm based cognitive radio. In *2013 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–4, May 2013.
- [wim] wimax. <http://wimaxforum.org>. Accessed 22 Feb 2018.
- [xil] Xilinx corporation. <https://www.xilinx.com>. Accessed : 2018-09-03.
- [XLSSL14] Yubin Xu, Limin Li, Boon-Hee Soong, and Cheng Li. Fuzzy q-learning based vertical handoff control for vehicular heterogeneous wireless network. In *Communications (ICC), 2014 IEEE International Conference on*, pages 5653–5658. IEEE, 2014.
- [XPN15] Tian Xia, Jean-Christophe Prévotet, and Fabienne Nouvel. Microkernel dedicated for dynamic partial reconfiguration on arm-fpga platform. *SIGBED Rev.*, 11(4) :31–36, January 2015.
- [YZL11] J. Yan, L. Zhao, and J. Li. A prediction-based handover trigger time selection strategy in varying network overlapping environment. In *Vehicular Technology Conference (VTC Fall), 2011 IEEE*, pages 1–5. IEEE, 2011.
- [ZAT15] Aymen Ben Zineb, Mohamed Ayadi, and Sami Tabbane. Who decision using a fuzzy reverse mlp with reinforcement learning. In *Communications and Networking (COMNET), 2015 5th International Conference on*, pages 1–6. IEEE, 2015.

- [ZEA05] Yutian Zhao, A. T. Erdogan, and T. Arslan. A novel low-power reconfigurable fft processor. In *2005 IEEE International Symposium on Circuits and Systems*, pages 41–44 Vol. 1, May 2005.
- [ZKS07] Q. Zhang, A. B. J. Kokkeler, and G. J. M. Smit. An efficient fft for ofdm based cognitive radio on a reconfigurable architecture. In *2007 IEEE International Conference on Communications*, pages 6522–6526, June 2007.







## AVIS DU JURY SUR LA REPRODUCTION DE LA THESE SOUTENUE

**Titre de la thèse:**

Gestion de la reconfiguration dynamique dans un contexte de communication numérique sans fil

**Nom Prénom de l'auteur : RIHANI MOHAMAD AL FADL**

**Membres du jury :**

- Monsieur CHEHAB Ali
- Monsieur MROUE Mohamad
- Monsieur MOHANNA Yasser
- Monsieur PREVOTET Jean-Christophe
- Monsieur ROMAIN Olivier
- Monsieur BOURENNANE El-Bay
- Madame NOUVEL Fabienne
- Monsieur GRANADO Bertrand

**Président du jury :** B. Granado

**Date de la soutenance :** 18 Décembre 2018

Reproduction de la these soutenue

- Thèse pouvant être reproduite en l'état  
 Thèse pouvant être reproduite après corrections suggérées

Fait à Rennes, le 18 Décembre 2018

Signature du président de jury

Le Directeur,

M'hamed DRISSI



A handwritten signature in black ink, appearing to be "B. Granado", written over a horizontal line.

**Titre :** Gestion de la reconfiguration dynamique dans un contexte de communication numérique sans fil.

**Mots clés :** FPGA – Machine Learning – Radio Cognitive – Reconfiguration Partielle – Réseaux de Neurones – SoC

**Résumé :** Aujourd'hui, les appareils sans fil disposent généralement de plusieurs technologies d'accès radio (LTE, WiFi, WiMax, ...) pour gérer une grande variété de normes ou de technologies. Ces appareils doivent être suffisamment intelligents et autonomes pour atteindre un niveau de performance donné ou sélectionne automatiquement la meilleure technologie sans fil disponible en fonction de la disponibilité des normes. Du point de vue matériel, les périphériques System on Chip (SoC) intègrent des processeurs et des structures logiques FPGA sur la même puce avec une interconnexion rapide. Cela permet de concevoir des systèmes logiciels / matériels et de mettre en œuvre de nouvelles techniques et méthodologies qui améliorent considérablement les performances des systèmes de communication. Dans ces dispositifs, la reconfiguration partielle dynamique (DPR) constitue une technique bien connue pour reconfigurer seulement une zone spécifique dans le FPGA tandis que d'autres parties continuent à fonctionner indépendamment. Pour évaluer quand il est avantageux d'effectuer un DPR, des techniques adaptatives ont été proposées.

Ils consistent à reconfigurer automatiquement des parties du système en fonction de paramètres spécifiques. Dans cette thèse, un système de communication sans fil intelligent visant à implémenter un émetteur OFDM adaptatif et à effectuer un transfert vertical dans des réseaux hétérogènes est présenté. Une couche physique unifiée pour les réseaux WiFi-WiMax est également proposée. Un algorithme de transfert vertical intelligent (VHA) basé sur les réseaux neuronaux (NN) a été proposé pour sélectionner le meilleur standard sans fil disponible dans un réseau hétérogène. Le système a été implémenté et testé sur un ZedBoard équipé d'un Xilinx Zynq-7000-SoC. La performance du système est décrite et des résultats de simulation sont présentés afin de valider l'architecture proposée. Des mesures de puissance en temps réel ont été appliquées pour calculer l'énergie de surcharge pour l'opération de RP. De plus, des démonstrations ont été effectuées pour tester et valider le système mis en place.

**Title:** Management of Dynamic Reconfiguration in a Wireless Digital Communication Context.

**Keywords:** Cognitive Radio – FPGA – Machine Learning – Neural Networks – Partial Reconfiguration – SoC

**Abstract:** Today, wireless devices generally feature multiple radio access technologies (LTE, WiFi, WiMax, ...) to handle a rich variety of standards or technologies. These devices should be intelligent and autonomous enough in order to either reach a given level of performance or automatically select the best available wireless standard. On the hardware side, System on Chip (SoC) devices integrate processors and FPGA logic fabrics on the same chip with fast inter-connection. This allows designing Software/Hardware systems. In these devices, Dynamic Partial Reconfiguration (DPR) constitutes a well-known technique for reconfiguring only a specific area within the FPGA while other parts continue to operate independently. To evaluate when it is advantageous to perform DPR, adaptive techniques have been proposed. They consist in reconfiguring parts of the system automatically according to specific parameters. In this thesis, an intelligent wireless communication system aiming at implementing an adaptive OFDM based transmitter is presented.

An unified physical layer for WiFi-WiMax networks is also proposed. An intelligent Vertical Handover Algorithm (VHA) based on Neural Networks (NN) was proposed to select best available wireless standard in heterogeneous network. The system was implemented and tested on a ZedBoard which features a Xilinx Zynq-7000-SoC. The performance of the system is described and simulation results are presented in order to validate the proposed architecture. Real time power measurements have been applied to compute the overhead power for the PR operation. In addition demonstrations have been performed to test and validate the implemented system.