



Network on chip based multiprocessor system on chip for wireless software defined cognitive radio

Muhammad Imran Taj

► To cite this version:

Muhammad Imran Taj. Network on chip based multiprocessor system on chip for wireless software defined cognitive radio. Other. Université Paris-Est, 2011. English. NNT : 2011PEST1050 . pastel-00665786

HAL Id: pastel-00665786

<https://pastel.hal.science/pastel-00665786>

Submitted on 2 Feb 2012

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.

Doctorate Thesis / Thèse de doctorat

Université Paris-Est, ESIEE Paris, Laboratoire LIGM, École Doctorale Maths
STIC, Spécialité: Electronique, Optronique et Systèmes

Presented and defended publicly by / Présentée et soutenue publiquement par

Muhammad Imran TAJ

12th September 2011/ le 12 Septembre 2011

Title / Titre:

**Network on chip based Multiprocessor System
on Chip for Wireless Software Defined and
Cognitive Radios**

Thesis supervised by: / Thèse dirigée par:

Prof. M. Akil (LIGM, ESIEE Paris, France).

Prof. O. Hammami (LEI, ENSTA ParisTech, France).

Jury Composition :

Professeur Yves LOUËT (SUPELEC Rennes, France)	Rapporteur
Professeur Jiří MASOPUST (FEL Plzeň, République Tchèque)	Rapporteur
Professeur Geneviève BAUDOIN (ESYCOM, ESIEE Paris, France)	Examinatrice
Professeur Serge WEBER (Université Henri Poincare Nancy, France)	Examinateur
Professeur Mohamed AKIL (LIGM, ESIEE Paris, France)	Directeur
Professeur Omar HAMMAMI (LEI, ENSTA ParisTech, France)	Co-directeur

Acknowledgements

First of all, I acknowledge my deepest gratitude and appreciation to my supervisor Professor M.Akil for his guidance and persistent help throughout my doctoral studies. Your professional support helped me to identify and to understand actual research problems and to pursue their solutions. Thank you for your patience, for countless thoughtful long technical discussions and for your friendly attitude as it has greatly contributed to my professional and personal growth and made it possible to complete this work.

Thanks to the co-supervisor Professor Hammami for having introduced me to the academic team at ESIEE Paris. Without his initial support and advice, this work could not have started. He has played an invaluable part in both my graduate studies and PhD Work. My acknowledgements are equally due for Professor Huggins for his guidance about the technical management which I used throughout this thesis.

I am grateful to all members of the jury for taking their time reading this thesis. Professor Louet's excellent comments helped me to immensely improve the manuscript. The remarks of Professor Masopust also helped to increase the readability of this manuscript. Professor Baudoin, apart from being external examiner, has been very kind to me throughout the thesis. She sent me all the relevant passwords to access IEEE articles and other online libraries. She has created a very wonderful research environment at ESIEE, without which I would have never been able to make it up. My thanks are equally due for Professor Weber, who very kindly accepted to participate as an external examiner.

I wish to express my thanks to the administrative staff at both universities, ESIEE and ENSTA for their kind help and assistance during the course of this research.

My sincere thanks go to my dear colleagues and friends from both the laboratories in ESIEE and ENSTA. The last three years have been an unforgettable experience for me and it has been a great pleasure to spend these moments with them, either in the laboratory or simply around a table during the pause-café.

My best friends, Laurent Fontaine and Ali Iqbal, have always been there for me throughout this period. I am highly indebted to them for their indirect positive impact on this work. Laurent's parents, Dominique and Jean François, have been a constant source of support and encouragement. Last, but truly not least, I want to thank to my parents for being with me and for having taught me to live and to love the moments of life we have. The work done in this thesis is a result of their indirect efforts and sacrifices they have made during whole of my childhood and youth. It would have never been possible without their unconditional love, support and understanding.

Abstract

Software Defined Radio (SDR) and Cognitive Radio (CR) are entering mainstream. These high performance and high adaptability requiring devices with agile frequency operations hold promise to:

1. address the inconsistency between hardware and software advancements,
2. real time mode switching from one radio configuration to another and
3. efficient spectrum management in under-utilized spectrum bands.

Framed within this statement, in this thesis we have implemented a SDR waveform on 16 Processing Element (PE) Network on chip (NoC) based general purpose Multiprocessors System on chip (MPSoC), with access to four external DDR2 memory banks, which is implemented on a single chip Xilinx Virtex-4 FPGA. We shifted short term development of a waveform into software domain by designing an efficient parallelization and synchronization strategy for each waveform component, individually. We enhance our designed waveform functionality by proposing and implementing three Artificial Neural Networks Schemes: Self Organizing Maps, Linear Vector Quantization and Multi-Layer Perceptrons as effective techniques for reconfiguring CR Transceiver after recognizing the specific standard based on input parameters, pertaining to different layers, extracted from the signal. Our proposed adaptive solution switches to appropriate Artificial Neural Network, based on the features of input signal sensed. We designed an efficient synchronization and parallelization strategy to implement the Artificial Neural Networks based CR Transceiver Algorithms on the aforementioned MPSoC chip. The speed up we obtained for our SDR waveform and CR Transceiver algorithms demonstrated that the general purpose MPSoC devices are the most efficient answer to the acquisition challenge for major organizations that invest or plan to invest in SDR and CR based devices, thereby allowing us to avoid expensive hardware accelerators. We address the case of a complex signal composed of many modulated carriers by dividing the PEs in individual groups, thus received signal with more than one Standard is processed efficiently. We add further functionality in our designed Multi-standard CR Transceiver possessing SDR Waveform by proposing a new approach for radio spectrum management, perhaps the most important aspect of CR. We make our designed waveform Spectrum efficient by modelling the primary user signal Radio Frequency features as a Non-linear Autoregressive Exogenous (NARX) time series, which is then given as input to Elman Recurrent Neural Network that predicts the evolution of Radio Frequency Time Series to decide if the secondary user can exploit the Spectrum band. We exploit the inherent cyclostationary in primary signals for NARX Time Series Modeling of Radio Frequency features, as predicting one RF feature needs the previous knowledge of other relevant RF features. We observe a similar trend between predicted and actual values.

Ensemble, our designed Spectrum Efficient SDR waveform with a Universal Multi-standard Transceiver answers the SDR and CR performance requirements under resource constraints by efficient algorithm design and implementation using lateral thinking that seeks a greater cross-domain interaction.

Key-words: SDR waveform, CR Multi-standard Transceiver, MPSoC, Artificial Neural Networks, Spectrum Evolution Prediction.

Abstract (in French)

La Radio Logicielle (SDR : Software Defined Radio) et la Radio Cognitive (CR : Cognitive Radio) deviennent d'un usage courant car elles répondent à plusieurs enjeux technico-économiques majeurs dans le domaine des télécommunications. Ces systèmes radio permettent de combler l'écart de développement technologique qui existe entre la partie matérielle et la partie logicielle des systèmes de communication, en permettant la gestion optimale des bandes de fréquences sous-utilisées par la commutation en temps réel d'une configuration radio à une autre.

Dans ce cadre, cette thèse présente la mise en œuvre d'une chaîne de traitements Radio Logicielle (appelé SDR waveform) dans un Système Multiprocesseurs sur Puce (MPSoC) à usage général (implémenté dans un FPGA de type Xilinx Virtex-4). Cette plate forme est basée autour d'un Réseau sur Puce (NoC) interconnectant 16 processeurs élémentaires (appelés PE) disposant de quatre blocs-mémoires externes DDR2. Nous avons proposé des implémentations temps réel et embarquées sur MPSoC de différentes briques de traitements d'une chaîne SDR, en concevant une stratégie efficace de parallélisation et de synchronisation pour chaque composante élémentaire de la « waveform ». Nous avons amélioré la fonctionnalité de la chaîne de traitement Radio Logicielle, en intégrant un Transceiver reconfigurable basé sur différents modèles de Réseaux de Neurones Artificiels (RNA) : les Cartes Auto-Organisatrices (SOM), les Réseaux de Neurones Compétitifs (LVQ) et enfin les Réseau Multi-Couches de Perceptrons (MLP). Ces trois RNA permettent la reconnaissance de la norme spécifique basée sur les paramètres d'entrée extraits du signal et la reconfiguration du Transceiver de CR. La solution adaptative que nous avons proposée commute vers le RNA le plus approprié en fonction des caractéristiques du signal d'entrée détecté. Il est important de pouvoir prendre en compte des signaux complexes et multi-porteuses. Dans ce cadre, nous avons adressé le cas d'un signal complexe composé de plusieurs porteuses, ainsi en divisant les PEs en différents groupes indépendants, nous affectons chaque groupe de PEs au traitement d'une nouvelle porteuse. Nous avons conçu une stratégie efficace de synchronisation et de parallélisation de ces trois RNA pour CR Transceiver. Nous l'avons appliquée, par la suite pour l'implantation des nos algorithmes sur le MPSoC déjà cité.

L'accélération que nous obtenons pour la SDR waveform et pour les algorithmes de Transceiver de CR démontre que les MPSoC à usage général sont une réponse pertinente, entre autres, aux contraintes de performances sur une telle plate forme. Le système que nous proposons apporte une réponse aux défis technico-économiques des grandes entreprises qui investissent ou prévoient d'investir dans des équipements basés sur des SDR ou des CR, puisqu'il permet d'éviter de recourir à des équipements d'accélération coûteux.

Nous avons, par la suite, ajouté d'autres fonctionnalités à notre waveform avec un « CR Transceiver multinormes », en proposant une nouvelle approche pour la gestion du spectre radio. Ceci étant l'aspect le plus important de CR. Nous rendons ainsi notre waveform spectralement efficace en modélisant les caractéristiques radiofréquences (RF) du signal utilisateur primaire sous la forme d'une série temporelle multi-variées. Cette série temporelle est ensuite fournie comme entrée dans un Réseau de Neurones Récurent d'Elman (ERNN) qui prédit l'évolution de la série temporelle de RF pour déterminer si l'utilisateur secondaire peut exploiter la bande de fréquences. Nous avons exploité la cyclo-stationnarité inhérente des signaux primaires pour la Modélisation Non-Linéaire Autorégressive Exogène (NARX : Non-linear AutoRegressive Exogenous) des séries temporelles des caractéristiques RF, car la prédiction d'une caractéristique RF demande d'abord de connaître les autres caractéristiques radios pertinentes. Nous avons observé une tendance similaire pour les valeurs prédites et observées.

En résumé, nous avons proposé des algorithmes pour SDR waveform à efficacité spectrale avec un Transceiver Universel, ainsi que leurs implantations parallèles sur MPSoC. Notre conception de waveform répond aux exigences en performances et aux contraintes de ressources embarquées des applications dans le domaine.

Table of contents

List of Figures	14
List of Tables.....	17
Chapter 1: Motivation and Context.....	19
1. Introduction:	19
2. First Contribution [183, 184, 185]:.....	20
3. Second Contribution [186, 187]:	21
4. Third Contribution [188, 189]:	23
5. Conclusions and Thesis Organization:	24
Chapter 2: Software Defined Radio- State of the art.	26
1. Software Defined Radio-Definition:	26
1.1 Features of SDR:	30
1.1.1. Portability:	31
1.1.2. Reconfigurability:.....	31
1.1.3. Seamless Mobility:	31
1.1.4. Interoperability:	32
2. Issues in SDR:	32
2.1 Application Engineering:.....	33
2.2 Computational Capacity:	33
2.3 Analog to Digital and Digital to Analog Conversion:	34
2.4 Software Download:	35
2.5 Processing Elements Choice:	35
2.6 Standardization and Certification:	36
2.7 Smart Antenna Design:	37
3. SDR Architecture:	37
4. Software Communication Architecture:	40
5. Academic SDR Reference Implementations:	43
5.1 SCARI [18]:	43
5.2 OSSIE [7]:	45
5.2.1. Waveform Workshop – Tools of OSSIE [8]:	46
6. Commercial SDR [19]:	46
6.1 SDR-4000 [19]:	47

7. Conclusion:	49
Chapter 3: Cognitive Radio - State of the art:	51
1. Introduction:	51
1.1 Definition:	54
2. Theoretical Research Issues:	55
2.1 Sensing:	56
2.2 Decision making process:	56
2.3 Learning Process:	57
2.4 Architecture Implementation:	57
2.5 Equipment Test Procedures and Certifications:	58
2.6 Interaction with all Layers of Protocol Stack:	58
2.7 RF Design:	58
2.8 SoC Implementation:	59
2.9 Accurate and Secure Primary User Detection:	59
2.10 Interoperability:	60
2.11 Security:	60
3. Cognitive Radio Architecture:	61
4. Machine Learning:	63
4.1 Memorization:	64
4.2 Swarm Intelligence:	64
4.3 Genetic Algorithms:	65
4.4 Fuzzy Logic:	66
4.5 Other Methods:	67
4.6 Artificial Neural Networks:	68
5. Major Cognitive Radio Projects and Achievements:	70
5.1 IDROMeL [37]:	70
5.2 GRACE[38]:	72
5.3 End to End Reconfigurability (E2R) [39]:	72
6. Conclusions:	73
Chapter 4: Embedded System Implementation and Optimization of SDR.	75
1. Introduction:	75
2. SDR Embedded Implementation Efforts:	76
3. MPSoC and FPGA Technology:	78

4.	The Multiprocessor Platform:.....	79
4.1	External Constraints:	80
4.2	MPSoC Architecture:	81
4.3	Processing Element:	82
4.4	OCN:	83
5.	OSSIE Core Framework:.....	84
5.1	OSSIE Signal Processing Library:	85
6.	Performance Evaluation of OSSIE SigProc on Embedded Platform	86
	ML-403 Board:.....	86
6.1	Microblaze cache:.....	90
7.	OSSIE Signal Processing Function Performance Enhancements.....	92
	through Parallelization:	92
7.1	The Parallelization Primitives:	93
7.2	Parallelization Description:	94
8.	Additional Functionality in SDR waveform:.....	97
8.1	FFT Parallelization Strategy:.....	99
8.2	Viterbi Decoding Parallelization Strategy:.....	100
9.	Experimental Results for FFT and Viterbi decoding Parallelization:	102
10.	Other Functions to be realized:.....	103
11.	Conclusions:	105

Chapter 5: Standard Recognizing Artificial Neural Networks Based Reconfigurable Cognitive Radio Transceiver and its Embedded Implementation. 107

1.	Introduction:	107
2.	Neural Networks Review:	109
2.1	Structure and Architecture of Artificial Neural Networks: LVQ, SOM and MLP:	111
3.	Related Work:.....	112
4.	Our proposed Approach:	113
4.1	SOM, LVQ and MLP for CR Standard Identification:	114
4.2	SOM and LVQ Implementation:	117
4.3	MLP Implementation:	121
4.4	Standard Recognition Experimentation Results:.....	123
5.	Parallelization and Implementation Strategy:.....	128
5.1	SOM and LVQ Parallel Implementation:.....	128
5.2	MLP Parallel implementation:	130
6.	Speed up Results:	131

7. Conclusions:	133
-----------------------	-----

Chapter 6: Cognitive Radio Spectrum Evolution Prediction using Artificial Neural Networks based Time Series Modeling..... 135

1. Introduction:	135
2. Spectrum Opportunity Sensing Methods:	138
2.1 Matched Filter:	138
2.2 Energy Detection:	139
2.3 Other approaches:	140
2.4 Cyclostationary Detection:	141
3. Multivariate Time Series Prediction and Neural Networks:	143
4. Multivariate Time Series representation of received signal:	144
5. ERNN Implementation:	146
6. Experimental Results:	149
7. Comparison with other Prediction Techniques:	152
8. Conclusions:	154

Chapter 7: Final Word and Perspectives. 155

Publications	159
References	160

List of Figures

Fig. 1. The practical SDR Block Diagram at receiver side	28
Fig. 2. Functional Architecture of ETSI's [15] SDR Radio Computer	39
Fig. 3. Classes making up Software Communication Architecture(page 97 of Book at [181])	42
Fig. 4. The Domain Profile –XML Files	42
Fig. 5. SCARI example waveform.(taken from [18]).....	44
Fig. 6. OSSIE Block Diagram	45
Fig. 7. SDR-400 Block Diagram. (taken from[19])	49
Fig. 8. Cognition Cycle [23].....	53
Fig. 9. Reconfigurable Cognitive Radio Cycle.	55
Fig. 10. Genetic Algorithm Structure.	66
Fig. 11. Neural Network node illustration. Shown are the input vector A (with elements $a_0 \dots a_n$); weight matrix W (with elements $w_{0j} \dots w_{nj}$); bias value, b; and output	69
Fig. 12. Artificial Neural Network	69
Fig. 13. Hardware Architecture of IDROMeL [37]	71
Fig. 14. Baseband Processor Architecture of IDROMeL (taken from [37])	71
Fig. 15. ITRS Roadmap.....	78
Fig. 16. FPGA board	80
Fig. 17. MPSoC Target External Connections [61].....	80
Fig. 18. Block Diagram of designed MPSoC Architecture [61]	81
Fig. 19. Virtex-4 Internal Architecture.....	81
Fig. 20. Microblaze based Processing Element [61]	82
Fig. 21. Microblaze Core Block Diagram	83
Fig. 22. Resource use in percentage.[61]	84
Fig. 23. Block Diagram of ML403.....	88
Fig. 24. Detailed Description of Virtex-4 ML403 Evaluation Platform Components (Front View)	88
Fig. 25. Detailed Description of Virtex-4 ML403 Evaluation Platform Components (Back View).....	88
Fig. 26. Data Cache Organization for Microblaze.....	90
Fig. 27. Block Diagram for Random Data Transceiver.....	92
Fig. 28. Parallelization Strategy for OSSIE Filter Functions	93
Fig. 29. CalculateRRCFilterCoefficients Parallelized Algorithm Design.....	95
Fig. 30. DesignRRCFilter Parallelized Algorithm Design	96
Fig. 31. CalculateDerivativeFilterCoefficients Parallelized Algorithm Design	97
Fig. 32. Phase 1 of FFT Algorithm: Iterative Computation for each PE without communication.....	99

Fig. 33. Phase 2 of FFT algorithm: Communication between pairs of processors.....	100
Fig. 34. Processing done by each PE for Viterbi Decoding Algorithm.....	101
Fig. 35. Speed-up versus PEs for FFT and Viterbi Decoding	103
Fig. 36. Neural Networks Selection based on received signal extracted parameters.	114
Fig. 37. SOM and LVQ and MLP based Cognition Cycle Receiver.....	117
Fig. 38. SOM Algorithm for BMU (Standard) determination and weight update.	118
Fig. 39. Iterative MLP Algorithm for Standard and transmission power determination and weight update.	122
Fig. 40. Output grid showing different standards forming a cluster.....	124
Fig. 41. U-matrix for normalized parameters for all Standards.....	125
Fig. 42. U-matrix row showing equal distance between three map units of a cluster comprising of 802.11.g Standard.....	125
Fig. 43. MSE after 200 training epochs.....	126
Fig. 44. MSE after 700 training epochs.....	126
Fig. 45. LVQ and SOM parallelized Algorithm.....	129
Fig. 46. MLP Parallelized Algorithm.....	130
Fig. 47. Parallelization illustration with the primitives.	131
Fig. 48. Speed-up obtained for all three Neural Networks based cognition algorithms.....	133
Fig. 49. Recent Spectrum Utilization in Percentage in Europe (taken from [136]).	136
Fig. 50. Opportunistic Spectrum access by cognitive users.	137
Fig. 51. Block diagram of (a) the basic energy detector and (b) an energy detector in frequency domain.	139
Fig. 52. Block Diagram of Spectrum Sensing predicate decision.	145
Fig. 53. Elman Recurrent Neural Network.....	147
Fig. 54. Our designed ERNN and LM based Algorithm for Spectrum prediction in form of time series.	150
Fig. 55. Observed and Predicted values of time series over testing samples.	152
Fig. 56. Prediction error between predicted and observed time series.	152

List of Tables

Tab. 1. Radio Computer Framework Components and their functionalities	40
Tab. 2. Important Functions of SigProc namespace	87
Tab. 3. Worst case execution time of SigProc Functions	89
Tab. 4. Filter Functions Performance Comparison for different Cache sizes.	91
Tab. 5. Our Proposed Cache Size for each Filter Function	91
Tab. 6. Component Processor usage.....	92
Tab. 7. Speed up of OSSIE Filter Functions using Multi-processor platform	97
Tab. 8. FFT Speed up for different number of PEs.	102
Tab. 9. Viterbi decoding Speed up for different number of PEs.	102
Tab. 10. Standards identified by respective Neural Network Techniques.	115
Tab. 11. Parameters of Standards recognized using LVQ.....	115
Tab. 12. Parameters of Standards recognised using SOM	116
Tab. 13. Parameters of Standard recognized using MLP	116
Tab. 14. Comparison with other NN based CR proposals.....	127
Tab. 15. Speed-up of ANN Cognitive Algorithms using Multiprocessor platform.	132
Tab. 16. Spectrum Sensing methods and their limitations.	140
Tab. 17. Algorithm Parameters.	150

Chapter 1:

Motivation and Context.

1. Introduction:

Future mobile networks offer new opportunities to access information in an anytime, anywhere paradigm. Next wireless terminals will have to integrate an increasing number of functionalities to allow an increased interactivity with the surrounding environment itself enhanced with the paradigm of “Ambient Intelligence”. In this respect, the trends taken by mobile communications towards Software Defined Radio (SDR), Reconfigurable Terminals and Cognitive Radio(CR) create not only new opportunities but also new constraints, in already constraint environment, in terms of platforms optimization answering requirements (criteria) of performance, energy consumption and chip area. The emergence of the concept of Software Defined Radio and Cognitive Radio is very attractive for future mobile communications since it naturally matches reconfigurability of wireless and multimodes of operations. This has a direct influence on the design of new solutions oriented towards FPGA based reconfigurable platforms.

Software Defined Radio and Cognitive Radio Technologies are entering mainstream. This can be witnessed by observing a huge recent interest in Software Radio related wireless communications and different proposals to incorporate cognition in the software radio. A massive amount of literature has been published on the topic. The academia and industry both are eager to contribute to the research policy and regulation issues. Despite being introduced for the first time in 1991 by J.Mitola, this topic still happens to be very fresh and fascinating research interest as there are many technical questions still need to be answered, e.g. how to address the required high performance and high adaptability with respect to ITRS Roadmap Prediction? How to address the real time constraint mode switching from one radio configuration to another? How to configure the Cognitive Radio Transceiver? How to perform Spectrum Sensing based on previous learning experiences?

This thesis answers all these technical challenges and questions associated with this SDR and CR concept. We contribute towards the development of this technology in three fold way:

1. Efficient implementation of SDR based algorithms, comprising a complete SDR waveform, on embedded Network on chip based multiprocessor system on chip, together with appropriate

middleware evaluation.

2. Cognition incorporation in our designed Software Defined Radio waveform by Artificial Neural Networks based Multi-standard CR Transceiver design so as to enhance its functionality. This resulted in an Universal Transceiver capable of operating in different Standards.

3. Incorporating radio resource (spectrum) efficiency by predicting the future occupancy status based on previous learnt experiences, using Artificial Neural Networks. This resulted in efficient cognitive user decision to exploit the available spectrum opportunity, thereby optimizing the most valuable resource (Spectrum) usage.

We explain all these points hereafter:

2. First Contribution [183, 184, 185]:

Due to the relatively complexity of applications related to software radio and if one wishes to exploit at best the dynamic behavior of those applications in order to optimize their implementations, it is necessary to rely on a rigorous and complete system level design methodology. Multiprocessor System on chip (MPSoC) are prime candidates for the implementation of next generation wireless systems due to their strong computational potential. Although front end analog part issues are essential and are being currently tackled, numerous issues in the digital part are emerging. In particular MPSoC requires efficient QoS based inter-processor interconnections which are implemented with Network on chip (NoC) at our targeted platform. These NoC adapt to workload variations in the operating wireless environments and make the utmost use of available resources. We explore the potentials of reconfigurable NoC technologies through eFPGA for SDR and CR. Hence the first objective of this thesis is to propose new design methodologies and new architectures for Network on chip based multiprocessor SoC which are efficient with regard to the above mentioned criteria for SDR. We started our work by the performance evaluation of the only open source C based SDR Open Source SCA Implementation:: Embedded (OSSIE) developed at Virginia Tech University in the USA. OSSIE implements the radio-communication algorithms under the name space of *SigProc* (Signal processing Library). We used Xilinx ML-403 platform based on Virtex-4 FPGA and the softcore Microblaze processor for identifying the functions that need to be optimized. We ported the following four classes of algorithms that were part of SigProc namespace:

1. Filter Functions.
2. Algebraic Functions.
3. Modulation Functions.
4. Demodulation Functions.

These four classes of algorithms constitute a basic SDR waveform and so by porting all these algorithms on Microblaze softcore processor, we evaluated the performance of SDR waveform components on embedded platform to meet good area performance tradeoffs. As a result of this mapping, the filter functions were identified to be strong candidates for optimization. We proposed the appropriate cache size for each filter function in question. We further went a step ahead by parallelizing these filter functions to address the prediction of International Technology Roadmap for Semiconductors (ITRS) that out to 2017, software design productivity will fall behind hardware design productivity. This prediction has a direct impact on SDR and CR based platforms as major investments has been made to impart intelligent software defined behavior in these agile frequency based operational devices. To address this ITRS prediction with respect to SDR, we ported the identified optimization requiring filter functions onto a sixteen processing element (PE) Network on chip based multi-core single chip platform, that is developed on a Xilinx FPGA Virtex-4 FX based chip. Each processing element happens to be the same Microblaze softcore processor that we chose for ML-403 board. Then we proceeded to include further components in the waveform to change its basic nature into more advanced form by parallelizing the following two functions on the same 16 PE platform:

1. FFT.
2. Viterbi Decoding.

An advanced parallelization strategy was followed for these two waveform components this time. This is because of the inherent intercommunication nature of these two algorithms. Because of this very reason, the speed-up obtained was not linear this time, unlike our filter functions. We proposed enhancement methodologies to exploit our multiprocessor platform leading to further optimized results. So in our first contribution, we have ported and proposed to enhance all the components of a SDR waveform on our NoC based multiprocessor SoC architecture comprised of 16 PE, each element being Microblaze.

At the same time, we evaluated the five aspects of omniORB Object Request Broker, namely, Innovation Time, In Sequence, Out Sequence, Objects Registered and Multi-threading over a high speed network of multicore workstations. This contribution resulted in an efficient SDR waveform design on MPSoC to address efficiently, the high performance and high adaptability requirements of SDR.

3. Second Contribution [186, 187]:

We target our efforts toward a Multi-standard Cognitive Radio Transceiver design for our Software Defined Radio waveform. CR systems are based on SDR technology and utilize intelligent software

packages that enrich their transceivers with the ability to adjust their operating parameters, observe the results and eventually take actions, that is to say, decide to operate in a specific radio configuration, expecting to move the radio towards some optimized operational state. So, the aim of a CR is to self adapt to changing requirements from user's applications in order to provide QoS and self-management capability. Multimode reconfigurable devices that are able to adapt their behaviour to the environment, configuring themselves in an appropriate fashion are increasingly being adapted within the wireless industry. In this respect, future cognitive radio devices will have the capability to adjust their operating parameters to optimize the radio state after sensing the environment variables and estimating the channel state. This is visualized in cognition cycle. In this respect, we target our efforts in designing efficient algorithms that can address the multimode reconfigurable device requirements. We proposed Unsupervised Neural Networks, Self Organizing Maps (SOM) and Linear Vector Quantization (LVQ) and Supervised Neural Network, Multilayer Perceptron (MLP) to incorporate learning lessons from previous experiences, thereby recognizing a Standard amongst a predefined list of Standards, and in the case of huge data range, the estimated transmission power required, as well. We evaluated the feasibility of SOM, LVQ and MLP algorithms to address our cognitive needs in SDR and we obtained encouraging results and each standard in question was recognized by our proposed Neural Networks based cognition algorithms for CR, thereby enabling the radio to move in the recognized standard mode, and in the case of huge data range, estimating the transmission power required. This leads to optimize radio resource use (spectrum, battery, carrier frequency). The cognitive functionality is spread across the layers of communication architecture, resulting in coordination amongst the layers for an efficient use of available resources. The characteristics, to be identified, to switch the CR Transceiver in the desired Standard mode, pertain to different layers of a communication system, thus we optimize the cognitive transceiver across different layers. We propose to perform the Standard detection by the identification of parameters specific to a particular standard and these parameters pertain to different layers of communication architecture, thus we target at cross-layer optimization of the CR system. After evaluating SOM, LVQ and MLP for cognition incorporation in our radio waveform, we come up with an efficient parallelization strategy for implementation of SOM, LVQ and MLP and then implemented our algorithms using our designed strategy on our target 16 PE NoC based multi-core MPSoC chip, to again address the ITRS Roadmap Prediction. Putting all the discussion of our second contribution, together, in this part we contributed by:

1. SOM, LVQ and MLP algorithms design to enhance our waveform's functionality.
2. Parallelized Implementation of SOM, LVQ and MLP on NoC based multicore, single chip Xilinx Virtex-4 FPGA.

This resulted in an efficient Cognitive Radio Transceiver design that recognized the standard based on parameters received as input to the Neural Network, moving the radio in the recognized standard mode

to make adequate decisions, thereby enhancing the waveform functionality addressed in the first contribution.

4. Third Contribution [188, 189]:

Our third contribution targets at making our radio waveform resource efficient. The most important radio resource in today's spectrum constraint environment is Spectrum. WLAN 2.4 and 5 GHz bands are overpopulated as their capacity is small enough for a very high number of interested parties. Paradoxically, with a keen observation at any recent spectrum utilization measurement we will witness a gigantic asymmetry in spectrum usage. That is, even if the popular spectrum bands, like WLAN are highly overcrowded in many geographical areas, majority of spectrum bands, assigned to different systems are practically silent. They are not exploited at the fullest. This Underutilization is notably visible in the licensed bands. This is where the waveform spectrum efficiency comes into play, to overcome this paradox of spectrum allocation. In this contribution, we address this very spectrum scarcity issue, as only a small portion of the allocated spectrum is used everywhere and at all times. We propose an efficient methodology to alleviate this inefficient use of the spectrum. We exploit the cyclostationary feature detection to propose an efficient spectrum management algorithm using Elman Recurrent Neural Network (ERNN). We model the licensed signal Radio Frequency (RF) features as a NARX time series, which is then given as input to ERNN. The interdependence between different variables (RF features of primary user in our case) makes it more difficult to predict the time series with multiple variables. Therefore, majority of the modelling and predicting methods focus on the time series with a single variable. At the same time, single variable time series are known not to contain enough information to predict accurately the future instances. The prediction results are more reliable and accurate when there is adequate available information. Non-linear multivariate time series prediction is based on ample available information. Our motivation to choose NARX model of multivariate time series for Spectrum Evolution prediction comes from the fact that this time series model is known to have more information than its counterpart univariate time series models. It is because of this very reason that it is better to predict the primary user presence or absence using multivariate time series. With the ability to accurately predict any universal non-linear function, ERNN is a far better choice for non-linear modelling and prediction, than other available tools, such as ARMA, ARIMA and Markov Models. Thus, the requirement of avoiding interference to potential primary users in their vicinity is addressed in this contribution, using all this background. We have used the Non-linear Autoregressive Exogenous Time Series Model using ERNN to perform single step ahead prediction using second order cyclostationarity. We predict the presence of primary user based on the previous observations, which are modelled as RF time series.

This resulted in a similar trend between predicted and observed values of primary user Spectrum usage, thereby enabling CR to exploit expected available spectrum opportunities, in near future. This

contribution added Spectrum efficient utilization, in the designed waveform of first and second contribution.

5. Conclusions and Thesis Organization:

In this thesis, we address and propose solutions to the most important issues related to SDR and CR Technology that are thought to be the big bang of today's wireless communication industry. This Chapter described our motivation to pursue this challenging subject and introduced the important notions, together with summarizing our contributions towards the subject.

This doctoral thesis is divided into six more Chapters (Chapter 2 to Chapter 7). The rest of the thesis is organized as follows: Chapter 2 describes in detail the SDR concept and the motivation that leads towards this solution. We define the definitions and terminologies with which we stick throughout the thesis, notably the definition of SDR as defined by Wireless Innovation Forum. The Chapter introduces the functionality of an ideal SDR and the key features: Portability, Reconfigurability, Seamless Mobility and Interoperability, that it should possess. We follow this up with the explanation of the practical version of Software Radio, called Software Defined Radio as it is not possible yet to sample directly from antenna. The key research issues that need to be addressed to make this technology viable together with the architecture evolution are reviewed. The first ever SDR project, in recent times, leading to Software Communication Architecture (SCA) definition is described, together with the Core Framework concept. Finally, we describe two academic SDR examples: SCARI and OSSIE, and one commercial product example: SDR-4000 that successfully employs this technology.

Chapter 3 proceeds to the concept of Cognitive Radio (CR) with its fundamental entities, benefits and implications. We stick with the Wireless Innovation Forum's definition of CR throughout this thesis, so we define this very definition in this chapter. Theoretical research issues that need to realize this technology are divided into five groups and are reviewed in this Chapter. We give an overview of CR architecture evolution in terms of RF front-end design and implementation issues. Different machine learning techniques used for CR engineering are reviewed with a special emphasis on Artificial Neural Networks (ANN). We conclude this Chapter by having an overlook of important European research projects addressing the CR engineering.

In Chapter 4, we describe our designed SDR waveform in detail. We begin with the important SDR Embedded Implementation efforts, then we describe in detail our target MPSoC platform and the Processing Element (PE) used. The OSSIE Signal Processing Library is studied in detail in this Chapter and cache size optimization for the considered Processing Element is addressed. We start by the OSSIE Signal Processing Functions (Filter, Algebraic, Modulation and Demodulation) embedded implementation to identify the Filter functions as parallelization requiring functions. We describe the

parallelization strategy to enhance the OSSIE *SigProc* functions (DesignRRCFilter, CalculateRRCFilterCoefficients and CalculateDerivativeFilterCoefficients) through parallelization on our target MPSoC. We conclude the Chapter 4 by describing the two additional functions (FFT and Viterbi Decoding) parallelization strategy to port them on the same MPSoC. The Speed-up, of all the parallelized functions that constitute a SDR waveform, is analyzed in detail to conclude the suitability of the General purpose MPSoC based solution for SDR.

Chapter 5 describes the Universal Multi-standard CR Transceiver Design for our designed waveform explained in Chapter 4. We describe the architecture of the three Artificial Neural Networks: LVQ, SOM and MLP and then exploit them for the Standard recognition among a predefined list of Standards. We describe in detail our proposed Neural Networks algorithms for Universal Multi-standard Transceiver Design and compare them with the other proposals. Our efforts to design a Multi-standard ANN based CR Transceiver answers the question of rapidly evolving new Telecommunication Standards with each day, thereby forcing us to rethink about the current transceiver architecture. In the last part of this Chapter, we design an efficient parallelization strategy to implement the three proposed Transceiver Algorithms: LVQ, SOM and MLP, on our target MPSoC and finally we conclude the Chapter with Speed-up Evaluation.

Chapter 6 incorporates Radio Resource: Spectrum, efficiency by predicting Spectrum Evolution occupation in time. We begin this Chapter by recent spectrum utilization review in Europe. We give an overview of the existing Spectrum sensing methods, with an emphasis on Cyclostationary feature detection. We follow this up with Non-linear Autoregressive Exogenous Multivariate time series modeling using Neural Networks (Elman Recurrent Neural Network) and representation of the received signal using time series analysis. We explain our ERNN design and Levenberg-Marquardt Learning Algorithm used to predict the spectrum evolution in time. We conclude this Chapter by analyzing the experimental results.

The last Chapter (Chapter 7) summarizes and concludes major achievements of this doctoral thesis. We stress the importance of lateral thinking to gain a greater cross domain interaction as we have done in this thesis. We also present multiple possible future research directions and open problems that still need to be addressed. Towards the end of thesis, we list the publications that we have done and finally, we give the bibliography.

Chapter 2:

Software Defined Radio- State of the art.

1. Software Defined Radio-Definition:

We are living in a rapid pace of communication technology, which makes the communication devices out-dated soon after their engineering. To go with this pace, communication systems require transparent insertion of the latest technological communication devices. With the insertion of latest technology in the communication devices, the upgraded modern device should be able to communicate with legacy devices as well. The Software Radio Technology allows one to add new functionality without hardware changes, even during a technological update. The ideal Software Radio, as defined by Wireless Innovation Forum [1] refers to the complete software control of the entire system. This means that analogue conversion should take place only at antenna, ensuring the support for a wide frequency range. These kinds of software radios will be obviously future proof as the whole radio system will be dependent on programmability, leading to the same hardware behaving differently at different instances. Furthermore, in the paradigm of Ambient Intelligence, these days even a simple workstation has the possibility of integrating a 2G/3G card. This trend is further evolving to 4G with higher QoS by means of improved channel equalization techniques, smarter antennas engineering and ever enhancing protection coding methods. The different types of applications and usages demand different standards in wireless communication systems. Although all these systems have almost similar components, the ways these components behave differ greatly from standard to standard. Also, while migrating from one generation to next, wireless network operators face problems as the newer handsets may not be compatible with newer generation network. In this regard, a reconfigurable or reprogrammable radio is required that can show different functionality with the same hardware. As defined above, the future proof *Software Radio* answers this requirement by sampling the antenna output directly, which is an impossible task as some RF front end, e.g. for amplifying and filtering, cannot be avoided to be performed at Analogue Front End. Nevertheless, this notion is different to the previous and traditional approach when the transceivers were hardware based. In that case, after the device engineering, it was exploitable only for the purpose for which it was designed. However, this approach was not able to answer the ever changing requirements of transceiver. The basic notion

behind Software Defined Radio, which is the realizable version of Software Radio, concept is the exact opposite of this very approach, i.e. as many functional blocks (ideally all: which is the definition of Software Radio) are software based as possible. This way, the functions that were carried out in hardware are performed by software, thus modifiable. Traditional Radios were built only for a particular frequency range, modulation type, and output power, which is in complete contrast to this SDR technology. This task is achieved by updating the software on the transceiver. The term was first coined in 1991 by J.Mitola [2].

The term Software Defined Radio (SDR) is the pragmatic version of the ideal Software Radio defined above. There are many different definitions of SDR, but we will stick to the definition of SDR Forum (Wireless Innovation Forum) [1] formed together with Institute of Electrical and Electronic Engineers (IEEE) P1900.1 group that says,

“a radio in which some or all of the physical layer functions are software defined.”

The physical layer functions refer to the specialized functional blocks that constitute the very functionality of SDR. This definition supports the notion of replacing as many hardware components as possible from traditional hardware based design to the software based SDR. As pointed out by Mitola,

“This term intends to signal the shift from digital radio to multiband multimode software-defined radios where “80%” of the functionality is provided in software, versus the “80%” hardware of the 1990’s. “

The very first definition covered only the range of operating frequencies from 2 MHz to 2 GHz. This first definition was meant for military applications as a single radio was desired to communicate with different radios that exploit different RF bands. With the commercial applications using this notion, this definition has evolved to a great extent with the passage of time. The ideal Software Radio places the A/D and D/A converters at the closest proximity of the antenna. This leads to signal processing functionality such as modulation at transmitter and tuning/detection of receiver signal and demodulation at the receiver being performed by different general purpose microprocessors and DSPs. Unfortunately, despite being the basic functional block of any SDR equipment, the performance advances in ADC and DAC are slower than other SDR functional blocks. This leads to the inability of ADCs and DACs to cope up with the high frequency signals. We have claimed the ADCs and DACs to be the basic functional block as they define the bandwidth and it is a well established fact that higher frequencies cause aliasing, leading to the requirement of anti-aliasing filter before ADC. It is because of this reason that the received signal is usually sampled at much higher sampling rates than required to relax the specifications of the anti-aliasing filter. The second implication of high bandwidth combined with dynamic range is the undesirable increase in power consumption. It has also been reported that the base stations and user terminals ask for more and more performance efficient

ADCs. Thus, the Software Radio is declined to Software Defined Radio: the pragmatic form of Software Radio, by the reduction of the bandwidth that has to be digitized by ADC. This is where the idea of Intermediate Frequency (IF) comes into play. This IF does not exist in the ideal Software Radio. However, in SDR, we are obliged to select a limited bandwidth from the full band. This practise leads to practical version of Software Radio: SDR, by bifurcating the whole communication waveform into analogue and digital parts i.e. the software processing is done at IF level. This is visualized in Figure 1. This also helps in overcoming the impossibility of wideband receiving antenna design that receives all multiband modes.

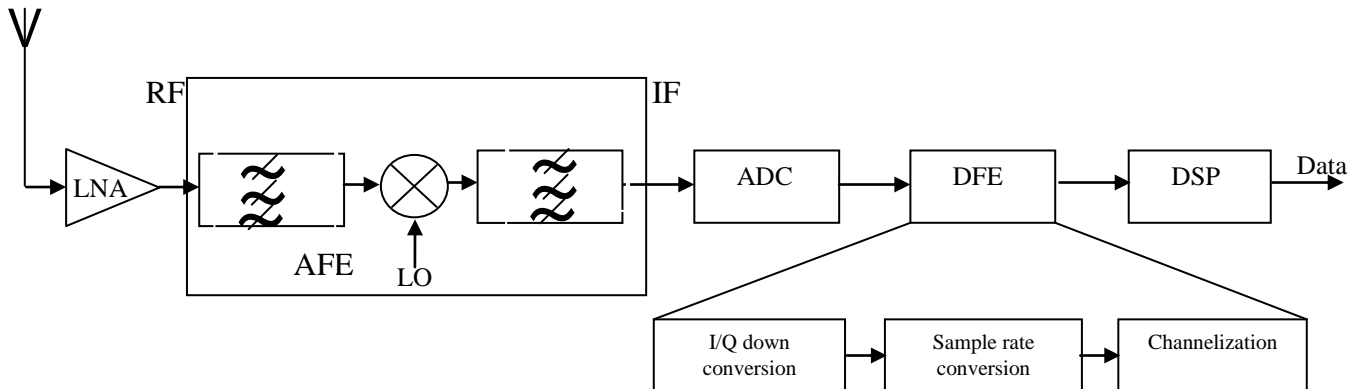


Fig. 1. The practical SDR Block Diagram at receiver side

We can see in Figure 1 that the functionality is divided into AFE and DFE. The AFE is necessary to select a bandwidth and to shift it from RF to an Intermediate stage, IF. The DFE is the real notion where the software part of the radio is implemented, i.e. analogue functionality is replaced by digital functionality. At the receiver side, this functionality comprises of I/Q down conversion, sample rate conversion and channelization, as shown in Figure 1. The channel selection is necessary in SDR as compared to ideal Software Radio because of the bandwidth selection criteria, due to the absence of ADCs and DACs supporting the huge bandwidth covering all the communication services required by the popular Standards. It includes baseband conversion and channel filtering. The most wireless standards cannot be implemented with 100 % software because of the reasons explained. The realization of ADCs with extremely high sample rates, with the ability to process in software the bandwidths of all ranges, immediately after the antenna will be a breakthrough in this technology. Also the absence of dealing with extreme dynamic range, which is the measure of the highest and lowest level signals present simultaneously in the radio, is another ADC/DAC related factor degrading the ideal Software Radio to Software Defined Radio. It is because of this reason that ADC conversion is not practical immediately after the antenna. Also, the signal processing with respect to DFE including channelization cannot be performed at RF level. However, in our opinion it will take several years to realize such ADCs with sufficient dynamic range, quantization and sampling frequency, that could support the bandwidths covering all the services provided by the terminal. We have already stated above that ADC engineering is less accelerated research area with respect to SDR engineering.

Software Engineering for SDR is a challenging task as it encounters the key software design concepts such as portability and reusability of the applications code. This challenge is further enhanced when seen in the framework of embedded systems, where very serious resource constraints are encountered. There has been an exponential growth in the ways and means of communication needs: data, video, voice command and control, emergency communication etc. – In this very paradigm modifying and updating the transceiver in a cost-efficient manner has become extremely important. SDR technology is promised to bring the desired flexibility, cost effectiveness and power to drive the radio engineering technology forward in the most technological efficient way. This technology efficiency refers to facilitating transition from dedicated to general-purpose hardware, thereby substituting hardware with software processing. The notion of SDR is an efficient combination of hardware and software in such a fashion that the physical layer functions are modifiable with the least possible changes in hardware. This task is accomplished by programmable processing technologies. This includes FPGA, DSP, GPP, Programmable SoC or other application specific processing entities. The discussion carried out so far can be summarized in three points: moving A/D as close to antenna as possible, replacing hardware with software processing using FPGAs, for example, and substituting dedicated hardware with flexible and general purpose components.

Another aspect refers to the ever increasing traffic rate but decreasing amounts of Spectrum. The Cognitive Radio notion, based on SDR, addresses this issue. So, another usage of SDR can be seen in the CR engineering. This too requires more sophisticated implementation of signal processing algorithms to be mapped on the radio. The need of deploying multiple standards within a single device can be addressed in the most efficient way by this sophisticated algorithm implementation. In a software radio, multiple waveforms can be implemented in a software, using the same hardware. This means that a single radio is meant to communicate with many others only with the change in software, whereas retaining the same hardware. This way interoperability is achieved within different communication needs defined above (data, video, voice command and control, emergency communication). This leads to the easy adaptation of new technologies at a reasonable cost. In order to engineer such radios, that are able to operate in many domains with the minimum or no change of hardware (ideally), there is a need of a standardized architecture. This architecture should help in achieving interoperability apart from reducing time to market by reducing the development time via component re-usage. The SDR provides a flexible radio architecture that allows changing the radio personality in time. This is referred to as Software Communication Architecture (SCA) [17]. The SCA is one of the key elements in the US military's Joint Tactical Radio System (JTRS), which is a SDR. We define SCA in detail in Section 4 of this Chapter.

SDR has found uses in academia, industry, government and military applications. As we have seen that Software Defined Radio has no single, unified, globally recognized definition, thus its definition constitutes the features and issues. It is only by understanding these issues that we can come up with

an efficient solution that addresses the wide range of applications and standards. The Section 2 of this Chapter provides an overview of all these aspects.

SDR is known to provide many opportunities in multiple domains, e.g. it replaces cellular base stations with Software Defined Multi-protocol base stations leading to the rapid introduction of the new Standards. At some point in time, it is also expected that SDR will provide a cheaper solution than conventional mobile terminals. Furthermore, it also gives an opportunity for anticipated wireless mobile users to personalize their units as Standards are upgraded. The concept of Cognitive Radio is also based on reconfigurability. In order to exploit the spectrum opportunities, CR needs platforms with faster reconfiguration capabilities. Chapter 3 is dedicated to the Cognitive Radio, however we need to make this a point that CR technology is based on SDR as well. In the long run, standardized open architectures are expected to become more popular. There are some disadvantages of SDR as well as in any real world entity. The security challenges are there as reconfigurable devices are more vulnerable to malicious attacks, e.g. during reconfigurability, there should be some mean to avoid the installation of malicious or altered code. The conventional security architectures are rigid enough to address this issue as they are known to be inherently less flexible. Also, the ideal Software Radio is still unrealizable as the non-practical goal of communication at any desired bandwidth, modulation and data rate is not achievable. Therefore, the researchers are working on the ADCs that will be capable of very high sampling rate with operating bandwidth of several GHz, greater quantization bits for dynamic range and low power consumption. When these goals will be achieved, a breakthrough can be expected in this technology. Nevertheless SDR has the potential to enhance the productivity of radio engineering by software.

We have noticed that SDR is a general device that can be reprogrammed to operate in various models, unlike AM or FM radios. The first ever device to demonstrate a completely software programmable radio was SPEAKeasy1[3]. Although it was a 6 feet tall rack of equipment but it led to project SPEAKeasy 2 that was a complete radio packaged in a practical radio size. It was the first SDR to include programmable vocoder and sufficient analog and DSP resources to handle many different kinds of waveform. We will explain more details about the SDR Engineering efforts in Chapter 4.

1.1 Features of SDR:

The SDR transceiver at both ends (transmitter and receiver) is required to possess some key features so as to enable the SDR to handle all the various broadcast standards – including the future expected standards, in this ever changing and adapting industry. These features are a major motivation within the commercial communications market. We define these key features in this section:

1.1.1. Portability:

A waveform portability refers to its movement from one platform to another without or minimal change in its components. Waveform applications should be able to rebuilt on another platform without rewriting the whole application. The Software Communication Architecture (SCA) contributes to this very feature. We define it in detail in Section 4 of this Chapter. However, even if a waveform is SCA compliant, it is not a sufficient condition that it will meet all the aspects of portability. The various components of a waveform mapped on FPGAs and DSPs require a transport mechanism. SCA 2.2.2 specifications [17] prescribe adapters between components. These adapters are primary means of communications and are between CORBA and FPGA components. The portability also requires that the component is translated correctly on the new platform. This means that the compatibility issues of language and target processor functionality should be addressed.

1.1.2. Reconfigurability:

This feature is related to the Portability that we have just defined. However, the only addition is that of dynamic reconfiguration of the waveform. The waveform should be reconfigured dynamically in such a way that the end user should not sense the internal change. However, different processing elements, in particular, different programmable processing technologies possess different features ultimately supporting different functionalities. Another important aspect is the trade-off between reconfigurability and energy efficiency in the embedded paradigm. The more energy efficient waveform component tends to be less reconfigurable and vice-versa. The interfacing of all the waveform components to constitute an application seems like a simple task, however it may lead to less reconfigurable waveform if not properly taken care of. The SDR should be able to process signals of all the Standards of specific application. This will enable the capability to change the functionality of radio during mission development or after the launch of mission.

1.1.3. Seamless Mobility:

The multiple analog transceivers approach is the equivalent of this feature in traditional radio engineering. Although the cellular services possess the feature of all time connectivity, the seamless mobility has not been realized 100 % till date. As an example, the wireless hotspots for 802.11 are yet not figured out easily. Thus the feature of seamless mobility leading to ubiquitous connectivity still needs further work out. In this regard, we have designed a Transceiver that we will explain in Chapter 5. The multiband solution is a way to achieve this seamless mobility. A multiband solution refers to a transceiver architecture that accommodates multiple bandwidths, signal levels and modulation types. We will explore all these issues in detail later in this thesis. A very simplistic approach to explain this ubiquitous connectivity is to opt one single Standard and attempt global regulation, thus providing an

idealistic vision. However, due to obvious reasons, this over simplistic approach is not technologically possible.

1.1.4. Interoperability:

The feature of interoperability has always been a challenge in public safety and military communications. It refers to the support of open architecture radio systems, e.g. Vanu Inc [4, 5] has demonstrated a multi-mode multi-band radio operating on a Compaq iPAQ platform. Robust methods for identifying incoming waveforms are the interoperability capabilities that are required for future SDRs. The SDR platform has an analog front end for signal transmission and a processing element to perform the modulation and demodulation functions in the radio. The modulation scheme used is normally known before hand. Interoperability seeks to devise solutions to dynamically identify waveforms by their analog and digital characteristics. The detection would then allow multiple radio platforms to communicate autonomously with each other. Core Framework concept, that we will explore in Section 4 of this Chapter is the way to achieve Interoperability in SDR engineering design.

Thus, SDR is a modern radio communication system with the addressed key-features that lead to software implementation of its components. The key features defined above make this a dominant technology in the long term. SDR has been deemed as the future of telecommunications, as most radio devices are expected to be SDRs in the near future. In this respect, this Section has started the Chapter 2 with a detailed introduction and defining features of SDR which are essential to understand.

2. Issues in SDR:

There are numerous issues and challenges that need to be addressed, so as to make this technology 100 % viable. The fundamental challenge still remains the same as we have discussed in the SDR definition, i.e. processing the waveform applications with multiple components with reasonable computational capacity, in such a way that the need of IF stage should be eliminated, or mimimized to a great extent. Also the discussed features in Section 1.1 should be incorporated. Starting from the very first reviews in this topic [6], there have been significant issues that have been brought to notice, such as issues in smart application engineering, computational requirements, analog to digital conversion, efficient and secure way to download software, processing elements choice, certification and standardization issues and smart antenna design. This Section defines all these issues. However, this list is, by no means, an exhaustive detail of all the research issues and challenges that need to be addressed.

2.1 Application Engineering:

The waveform components engineering is a difficult and challenging issue. The difficulty is further added when the SCA compliant application needs to be developed. Learning and comprehending SCA may prove challenging and consume several months. We started our work by Open Source SCA Implementation:: Embedded (OSSIE) [7,8] which we define later in this Chapter. OSSIE is an open source implementation that can be useful for beginners. There are certain aspects that need to be well understood while defining SDR waveform components manually, notably: CORBA, Object Oriented Programming, Embedded Systems constraints. OSSIE reduces the learning curve and hence the application development efforts. The SCA specifications ask the use of CORBA as a middleware platform. CORBA is known to consume a lot of memory and other resources, which is further exacerbated by the embedded resource constraint environment. This calls for the need of another available middleware, given the fact that CORBA is not a popular tool any more due to its tedious learning curve and less throughput, which is already dependent on message size to be transferred. On the other hand, the feature of portability is compromised. There have been some Object Request Brokers (ORBs) developed for FPGAs in recent years [9]. This is a way to achieve portability as the CORBA communication between all the processing elements will be easier. However, still the processing elements will encounter the latency and throughput implications, for which the CORBA is known for. The Data Distribution Service (DDS) is recommended as a substitute middleware to CORBA in the SCA framework. Till date, there has been no replacement of CORBA in the SCA context with another middleware meant to be less complex and better performing. There are also efforts going on to make ORBs faster and less memory hungry. In short, there exist ongoing Middleware activities because of which we can anticipate that soon there will be a clear path for middleware to make application engineering easier.

2.2 Computational Capacity:

A small hand held device is required to be multimode these days. In such multimode terminals, the basic issue is achieving the computational requirements in such a way that meet the power consumption, i.e. the power consumption should be below a certain threshold leading to the acceptable limits of battery discharge. At the same time, the temperature of the multimode handheld device should not be high enough to irritate the end user. SDR application engineering is addressed in the previous sub-section. This application performs signal processing at various stages for transmit and received signal, apart from other application control activities. We can divide the SDR waveform in waveform processing elements and administrative elements. Waveform processing elements have a defined algorithm. This is the reason that we can exploit the inherent parallelism in these components. This is an efficient way to address this issue of computational capacity. We delve into more details

about our efforts, in this regard, in Chapter 4. Administrative portion of the waveform has not that much level of inherent parallelism. The reason is that the administrative components are events driven, i.e. they are dependent on events to execute various administrative and control commands. In general the waveform components can be made to run in parallel. This can be explained by the distributed systems architecture support in SCA. The waveform processing elements require more computational capacity. It is because of this reason that we have addressed the waveform components parallelization strategy in Chapter 4 to address this issue.

2.3 Analog to Digital and Digital to Analog Conversion:

The issue of Analog to Digital Conversion is not only unique to SDR realization. However, it is meant to be one of the most important issues that can be used to significantly increase the flexibility of SDRs. The electronics converters are achieving more and more resolution together with faster conversion rates. The ideal case would be that the Analog to Digital Conversion should take place immediately at the Antenna. This will lead to the digital processing in software of all the radio communication functions, i.e. the need of any intermediate frequency stage will be eliminated. This task seems to be quite unrealistic at the moment, but research efforts are on their way in this very domain. This realization will lead to sampling being done immediately after antenna, thereby facilitating the signal processing portion of the waveform. The more sophisticated signal processing components of a waveform are heavily dependent on advances in this issue. The advances in SDR engineering is stimulated by ADC performance improvements, especially for sampling rates of approximately 100 million samples per second. The Nyquist Theorem states that the signals must be sampled at a rate of at least twice as that of highest frequency of the signal being sampled, leading to the accurate and faithful reproduction of the original signals from the samples. Echotek (a commercial SDR supplier) maintains an Analog to Digital Converter (ADC) or a Digital to Analog Converter (DAC) within a range of 0.5 dB of the specifications. The front ends are moving closer to the ADCs and the DACs are moving towards the transmitting portions, as the computation capacity issue is addressed. The performance of an ADC can be characterized by four parameters: stated resolution, signal to noise ratio, power dissipation and actual dynamic range. The most important aspect amongst these parameters is the stated resolution, which means resolution in bits for a given sample rate. The timing and synchronization to tackle a huge data rate in the whole SDR waveform is a difficult design issue, as well. The reason is that the information integrity throughout the radio waveform is compromised, if this aspect is under-estimated. Another important synchronization aspect is the timing issues between different processing entities, while each processing entity works at its own clock speed. This requires exact timing synchronization for each processing entity that is executing the specific waveform component.

2.4 Software Download:

The efficient methods for Software Download (load and install) are an essential requirement for the fully programmable implementation for multimode handheld devices. There are two methods, which can be used, namely: air interface download and smart card loading [6]. In the first method, an illegal intruder can obtain the software and alter it while in transport. This poses a serious security issue, especially in military and defense applications. This can be avoided by the use of efficient cryptographic algorithms. A Digital Certificate together with a public key is an efficient manner by which these kinds of malicious attacks can be avoided. There are many other authorization schemes that can be used to avoid any kind of illegal intrusion. In smart card loading, the software is downloaded by the insertion of a smart card in the terminal. This method has its own limitations, such as requirement of larger availability of sales points for such cards. Since the memory capacity has witnessed exponential growth, there is the possibility of having a priori information about all the existing standards that can be expected to be downloaded. Whenever the end user has the need of system change, the terminal only needs the proper identification of the new system, followed by the software download in a local fashion. The main objective of this issue is to find a reliable and real-time software download method that does not require any modifications on the existing user equipment, in an ideal case. There have been many propositions about an efficient software download strategy. However, still there is a need to address this issue as it has direct security implications, and is more vulnerable as compared to that of traditional radio.

2.5 Processing Elements Choice:

There has been a huge choice of processing element to be opted that could match with all the required features of a SDR. The static choice Application Specific Integrated Circuit (ASIC) is in contrast to the dynamic choice Reconfigurable Processing Element. There has been a very little effort for ASIC based processing element in the SDR waveform design due to the obvious reason of being non-reconfigurable. However, it can be useful in the cases where there is a preference for an efficient method of computing over portability and reconfigurability. Another possibility is the specialized co-processor design. These co-processors are specially designed for a specific waveform component such as FFT or Filter Function. These co-processors can be viewed as a compromise between ASIC and reconfigurable processing elements as they are not 100% rigid as ASICs, neither fully reconfigurable like the Microblaze nor other processing elements. They are given special parameters to change their behavior according to the specific waveform component that is required to be computed. The third option is the reconfigurable alternative to ASIC, FPGAs. To estimate the extent to which FPGAs can be reconfigured, we can see the compilers that have made it possible to generate FPGA code directly from MATLAB. However, this reconfigurability comes at the expense of power and area consumption. We can re-program the FPGA with the specific code needed for any waveform component with a

reconfiguration time as less as some fractions of second. This is extremely useful in the real time constraint environment where the end user requires ubiquitous connectivity. The other advantages of FPGAs include the rich tool sets and easily planned development tasks with moderate risks. Microprocessor systems also provide full real time programmability. Multiple core processors support to a great extent the parallelism required in SDR waveform components. There are two classifications of multi-core processors: Single Instruction Multiple Data (SIMD) and Multiple Instruction Multiple Data (MIMD). There is a single instruction stream in SIMD, i.e. each processor is constrained to execute the same instruction. It is useful in applications where each processor is required to operate on a different set of data. As an example of SIMD processing elements, we can quote SODA [10] and Sandbridge SB3011 [11]. In contrast to SIMD, MIMD have multiple instruction streams. This is the reason that separate programs can be executed over each other. It is considered as more flexible architecture than SIMD.

We have seen in this summary that there exists a huge choice between processing elements. The preference to a specific element is a personal choice which is geared by the priorities and trade-offs. The performance is not the sole criteria to measure a processing element. There are other features as well that need to be remembered, such as real time constraints, area and power constraints, cost and many such factors. An estimate of processing capacity of different processing elements can be done with the different measuring entities, such as MIPS and MFLOPS, but there is no single scale to have a 100 % correct estimate. We will give more details about our preference of processing element in Chapter 4 of this Thesis.

2.6 Standardization and Certification:

The basic standardization can be viewed in the framework of JTRS SCA specifications. Since 1996, Wireless Innovation Forum [1] comprising of academia and industry from all over the world has given a remarkable number of deliverables for commercial and defense sector. In Europe, The European Software Radio Architecture (ESRA) is an activity that aims at defining standards and certifications for SDR, in such a way that all the features are ensured in the waveform. A number of projects are aimed at this activity [12, 13]. The European Defense Agency has given an outline of the ESRA standardization activity [14]. The Wireless Innovation Forum and The European Software Radio Architecture are concerned with the standardization and certification of SDR. European Telecommunications Standards Institute (ETSI) [15] aims at defining the external interfaces between the base station and terminal. In USA, another standardization and certification effort is that of IEEE 1900.3[16].

2.7 Smart Antenna Design:

An antenna choice is a critical issue in SDR design as it supports multiple bands. The Wireless Innovation Forum defines a smart antenna system as: *A subsystem which includes the antenna that uses the spatial domain in combination with decision-based signal processing to improve link performance and enable other value added services* [1]. The design requirements for a smart antenna design is a multi-objective problem. A multiband antenna should serve all the different groups of frequency bands of different communication systems, in an ideal condition. The engineering of hand held smart devices require an antenna design which is thinner and more beautiful. The design of such antennas with little space in a complicated communication environment of today is a non-trivial task. The Planar Inverted F Antenna (PIFA) and The Folded Inverted Conformal Antenna (FICA) are commonly used in hand held devices. The common five communication bands: GSM850, GSM900, DCS, PCS and UMTS inter-switching should be supported in multiband reconfigurable antenna, within real time constraints. However, a specific antenna design works in different modes to support all the communication standards. The cross layered design approach for MIMO systems, to ensure smart antenna design for terminals and base stations, should be considered, thereby exploiting multipath in the propagation environment. The common performance metrics for antenna design: gain, polarization is not sufficient enough if the nature and degree of multipath are not given enough considerations. If capacity, the link and Media Access Control (MAC) layer characteristics are also considered, then the throughput of communication link can also be considered as an important performance metric.

In this Section, we gave an overview of the research issues that need to be addressed to make this technology 100 % viable. These research issues still need to be addressed and hence each one of them can be pursued as a separate Ph.D. topic. However, we have listed only the most important issues that are currently being tackled by the research community. These issues: smart application engineering, computational requirements, analog to digital conversion, efficient and secure way to download software, processing elements choice, certification and standardization issues and smart antenna design hold promise for improved performance, thus addressing the acquisition challenge which is posed by the realization of these devices.

3. SDR Architecture:

The progression in the radio generation leads to an increase in design rule, e.g. the functionality of early radios was limited to the transmission or reception of basic FM and AM signals, which is in extreme contrast to today's hand held smart devices. Early FDM radios had no software for physical

layer functions. 1990s saw a rapid growth of embedded processors of 8-bit and 16 bit in radios. Today such radios can be easily expected to have 10 k Lines of code (LOC) in many physical layer related functions. The adaption of SDR Architecture by the industry demands that the SDR Architecture should be capable of many hardware substitutes so as to always be ready to cope up with the never stopping continuously evolving new Standards. The organizations like Wireless Innovation Forum and European Telecommunications Standards Institute (ETSI) are on their way to define and develop reference architecture, e.g. in [15] there is a feasibility study that creates the ETSI SDR Reference Architecture Specifications for mobile devices, such as mobile phones as well as Reconfigurable Base Stations. The basic focus in the defined specification is on the flexibility to enable the device adaptation within its context. In this regard, the five basic mobile device requirements for SDR are considered in these specifications: Architectural, Capability, Operational, Interface and Miscellaneous. These requirements should be met to integrate the commercial consumer products like mobile phones and PDAs with radio platforms, containing some built-in radio functions, to give the architecture referred to as “radio computer”. This architectural proposition is meant to incorporate the highest level of flexibility in the SDR. Since a modular architecture supports radio applications from different providers, this is the very option in the architectural requirements. Also the key features of a SDR that we have discussed in Sub-section 1.1 are achieved by the modular architecture. A heterogeneous multi-processor architecture with inherent support for real time applications and that respects the defined power constraints is proposed. The architecture must support the dynamic reconfiguration of the radio platform that has to be integrated to form a radio computer. Security issues for the radio computer, in both licensed and unlicensed frequency bands should be supported. In order to support the architectural requirements, the interfaces in the SDR reference architecture are defined. A multi-radio SDR equipment should have such an architecture that may accommodate most of the available radio technologies, if not all, to satisfy the end user needs. The radio computer concept, in this regard, is important as it is actually an embedded system whose many parts are ASICs based, which are meant to control RF and DSP signal processing for baseband protocols, together with the relevant microprocessor running the relevant software. Radio Access Technology has advanced enough that SIMD and multi-core processors based architectures can be easily made to work with ASIC accelerators. ASICs, despite not being reconfigurable, are preferred in the waveform components where less power consumption is required, together with the functionality that has to be repeated over and over again. Thus, we can predict that the future SDR Architectures will be equipped with the qualities that will make them work more like a computer, where all the applications are software based and run on more general purpose computing elements. The most distinctive feature of such a radio computer will be the ability to change the running radio program and load new radio application software, even at run time, so that real time constraints are met. The modular architecture of the radio computer under discussion, as proposed by ETSI is shown in Figure 2. The services provided by the radio computer to its user application include positioning, connectivity and data transfer. The

administrative user has some additional privileged services such as installation and de-installation of new radio application programs. The responsibilities of the radio computer framework shown in Figure 2 are shown in Table 1. Table 1 is based on the functions described by the ETSI SDR Reference Architecture for Mobile devices. The reconfigurable and flexible RF circuitry is also one of the key features for the modular radio computer based SDR. We know that the ultimate

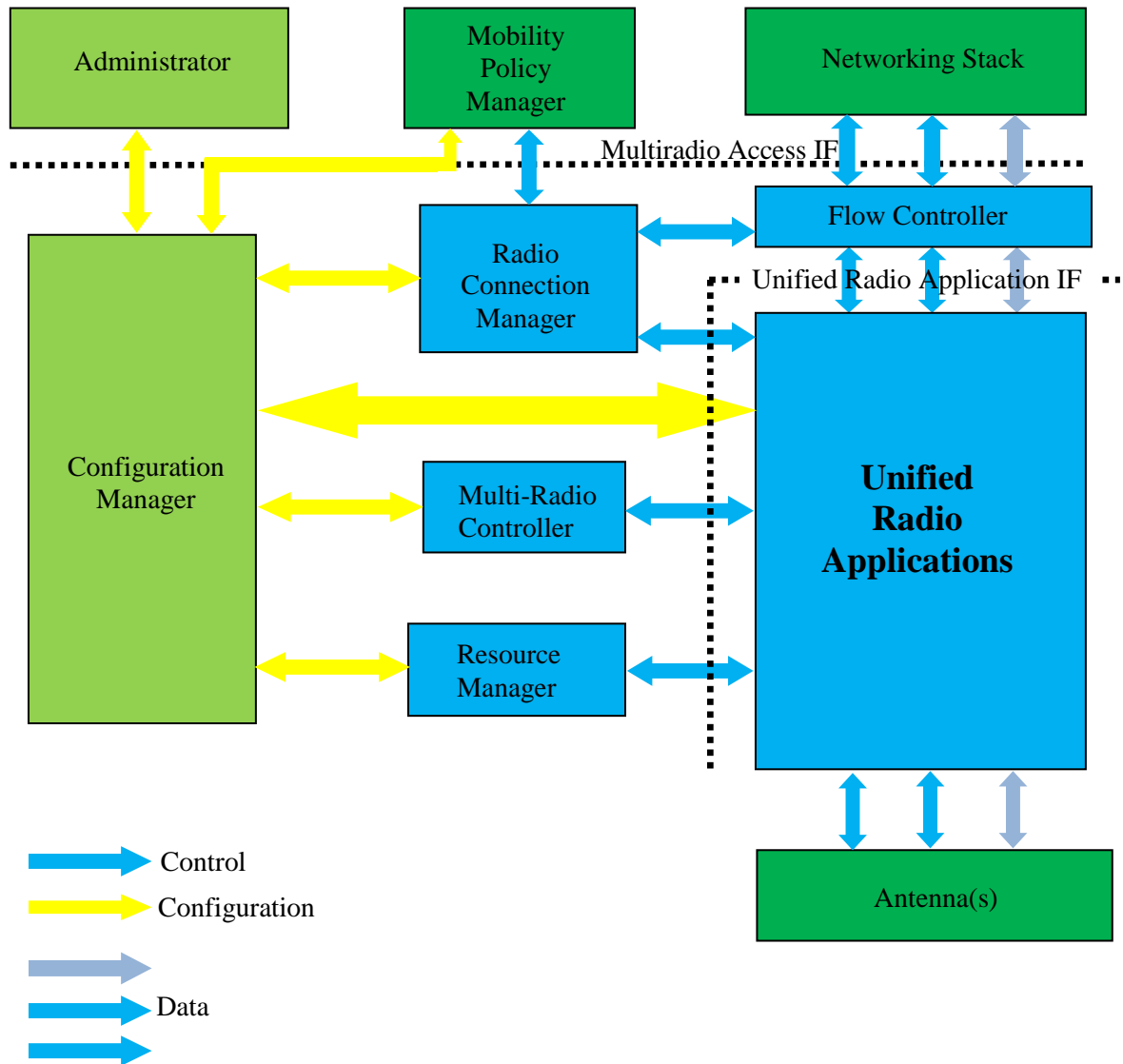


Fig. 2. Functional Architecture of ETSI's [15] SDR Radio Computer

goal of an universal RF circuitry is unrealizable, but at the same time the radio computer providing several RF circuits for different radio technologies will be in a better position to reconfigure and reuse the hardware peripherals. Apart from the detailed Architecture proposed by ETSI, that we have explained in this Section, there are other reference architectures of SDR and the relevant activities. We have already referenced the Wireless Innovation Forum and its partners in academia and industry. There are other bodies as well that aim at defining the radio specifications and architectures for SDR,

e.g. Object Management Group (OMG), exploits the Software Communication Architecture (SCA) that we will explain in the next Section of this Chapter for SDR engineering. The SCA based solution for SDR known as “Core Framework”, relies on CORBA implementation. The OMG’s set of specifications have the possibility to decouple the SDR reconfiguration model from the underlying CORBA implementation.

Component	Responsibility
Configuration Manager	Installation/deinstallation of radio applications into radio computer
	Management of radio parameters
Radio Connection Manager	Deactivation of radio applications according to user requests
	Overall management of user data flows, including inter-application switching
Flow Controller	Exchange of user data packets
	Flow Control
Multiradio Controller	Request scheduling on spectrum issued by simultaneous radio applications in order to anticipate interoperability problems.
Resource Manager	Management of radio computer resources
	Share the resources according to application priority
	Meeting real time application resource requiring constraints

Tab. 1. Radio Computer Framework Components and their functionalities

4. Software Communication Architecture:

We have already discussed the key features of the SDR in Sub-section 1.1. of this Chapter. The Joint Tactical Radio Systems (JTRS) is aimed at incorporating these features in quick and cost effective manner. It dates back to 1997. It was the first time that the waveform was defined as:

“Components of an Application that perform majority of the communications in a radio”[17]

The components addressing the majority of the communications can be interpreted as the multi-standard transceiver functionality. This functionality can be achieved by replacing existing separated systems with flexible architectures. These flexible architectures should be designed in such a way that they make the most of the existing common features of different Standards. In this respect, the ever varying communications platform challenges were meant to address by modular and scalable architecture. It must be noted that the ETSI SDR Reference Architecture shown in Figure 2 supports modular functionality as well. The Software Communication Architecture (SCA) is a common specification standard and component based software framework, built on JTRS, for SDR. It defines an Operating Environment (OE) being exploited by the JTRS radios to answer the portability and

interoperability features requirements for commercial and civilian standards together with military applications. The specifications for services and interfaces used by the applications are also provided. These interfaces are defined by Common Object Request Broker Architecture (CORBA) IDL. The graphical representation is done by using Unified Modelling Language (UML). The OE contains: the Core Framework (CF), the Domain Profile, the CORBA middleware, Application Environment Profile, the Device Drivers and the Real Time Operating System. The Operating System is required to provide the necessary interfaces defined in Application Environment Profile of SCA. Component based development is a key feature of the SCA architecture. This key feature is addressed by the “Core Framework (CF) concept”. This architectural concept is actually the core of SCA interfaces and profiles when it comes to managing, deploying, interconnecting and intercommunicating of software application components in distributed embedded environment. The SCA defines, in the CF, a set of interfaces that govern the deployment and management of waveforms and their components. The distributed environment uses CORBA for inter-communication. CORBA defines the interfaces and their operations in such a way that abstracts the underlying software layers from developer. These interfaces are grouped and defined as:

1. **Base Application Interfaces:** It provides the management and control interfaces for all system software components. It includes Port, LifeCycle, TestableObject, PortSupplier, PropertySet, Resource, ResourceFactory.
2. **Framework Control Interfaces:** It controls the instantiation, management and destruction/removal of software from the system. It includes Application, ApplicationFactory, DomainManager, Device, LoadableDevice, ExecutableDevice, AggregateDevice and DeviceManager.
3. **Framework Services Interface:** It is responsible for additional support functions and services. It includes File, FileSystem and FileManager.

Figure 3 shows the Software Communication Architecture with all these Interfaces. These interfaces allow developers to focus on application design by defining low level architectural details. The Domain Profile, shown in Figure 4, is used to describe the components in the system by deploying the SCA compliant hardware device and software component implementation into the CF domain through defining the component properties and interconnections, i.e. the way a waveform expresses its deployment requirement. The three elements of the Domain Profile: Device Configuration Descriptor (DCD), DomainManager Configuration Descriptor (DMD) and Software Assembly Descriptor (SAD) are the root profiles that drive the SCA. This task is done in the form of XML files. These XML files define: individual components of an application, their interconnection, their properties and the properties of hardware device abstractions. The CORBA middleware can be seen as a software bus for all the OE CORBA-capable components. The communication between distributed waveform components is location transparent.

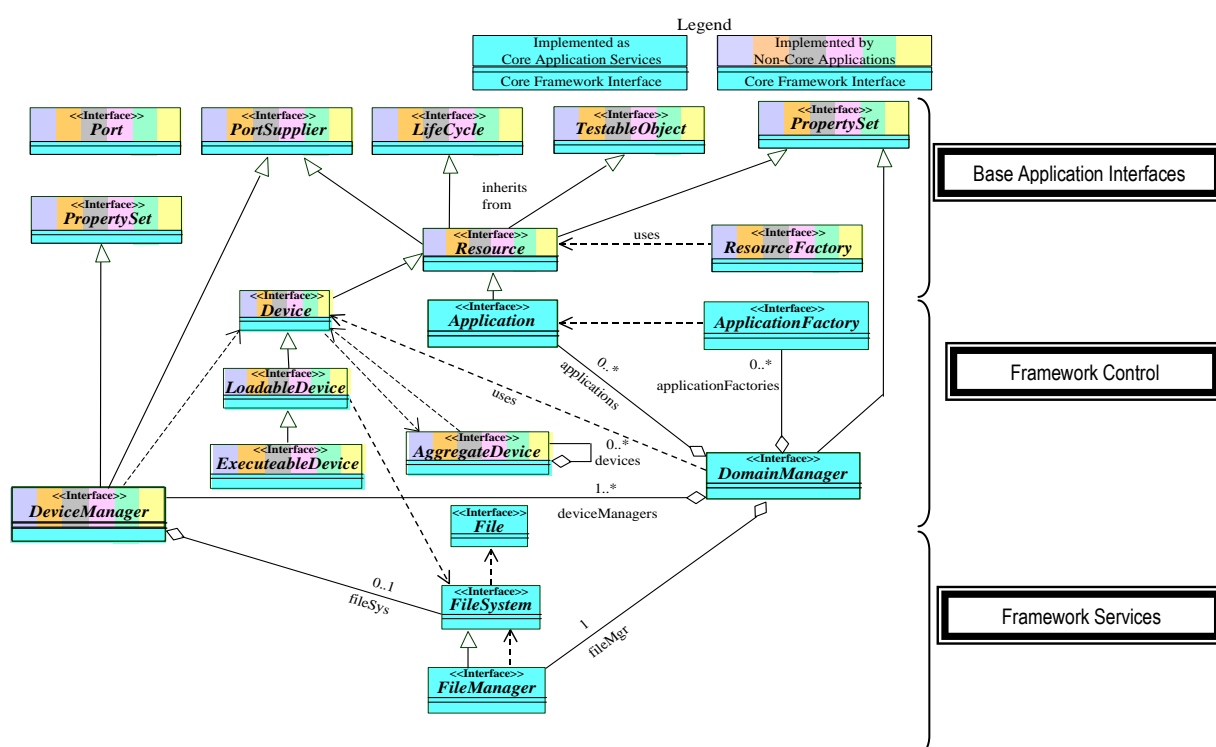


Fig. 3. Classes making up Software Communication Architecture(page 97 of Book at [181]).

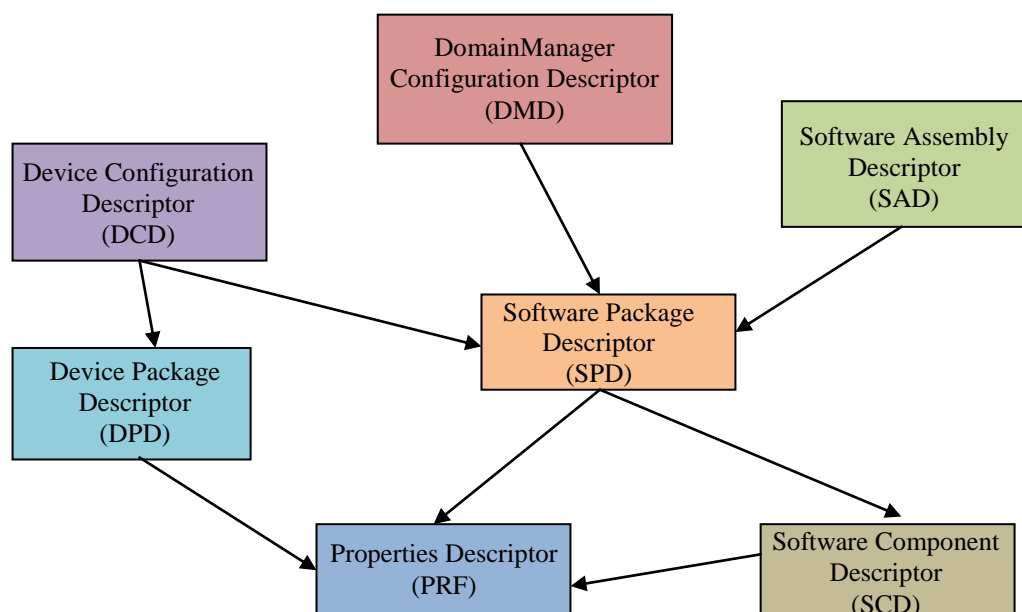


Fig. 4. The Domain Profile –XML Files

The JTRS SCA specifications are recent and immature because of which they go through continuous debugging, since the first version SCA 1.0 was released, to support the JTRS Clusters. SCA 3.0 was

released in 2004 but it is not supported on JTRS website. Also it required more work for an efficient implementation. It is a hunch in today's research community that the future SCA specification and core framework will evolve further. This may consist of smaller and simpler framework with less complexity and faster execution. Also the SCA should be re-structured to extend from current GPP environment to include the ORB usage in non-GPP processing elements. However, till date it remains the “de facto” software architecture for SDR Engineering, as it is considered the most widely used and referenced software architecture for SDR engineering as compared to its counterparts such as OMG's specification and GNU Radio Architecture. In future, it is anticipated that SCA will contribute a lot in increasing interoperability and reducing development cost.

5. Academic SDR Reference Implementations:

We have defined SCA in Section 4, which is implementation independent framework, thus subjected to numerous interpretations. However, for any framework, it is essential to be complemented by a Reference Implementation to define the behaviour of the specifications. A Reference Implementation is an open source implementation of any Standard designed by a trusted organization. The main objective is to make the Standard more comprehensible and encouraging the people to use it. Any wording in the Standard can be understood by the Reference Implementation of the source code. In this Section, we will examine the Reference Implementations of SCA. We examine the two academic JTRS-SCA containers below:

5.1 SCARI [18]:

The Software Communications Architecture Reference Implementation (SCARI) is a reference implementation of the JPEO SCA CF, developed by Communications Research Centre (CRC) Canada. Till date, it is treated as the de facto SCA CF. It is well documented and so more popular, as the most important goal of the Reference Implementation, clarity, is met. The full suite comprises of: SCARI++ and Application Toolset. SCARI++ has all the characteristics of CF that we defined in the previous Section. The Application Toolset assists in waveform development at all stages. Using it, we can produce waveform components and then assemble them to get a complete waveform. The SCARI suite makes it easier to develop a waveform by making different stages independent in SCA development lifecycle. The stages are: component modelling, component assembling and behaviour implementation in SCA environment, component deployment and finally execution. The components are created, assembled and then deployed. The functional requirements of the components are analyzed before creating them. This helps in preparing requirements specification. This specification is then used as input to determine the inner architecture of the components and the ports. We follow this

up by modelling the components by adding properties, component type and component. The properties and ports are small. Now, the source code can be generated and any specific algorithm e.g. FM waveform treatment can be added. The last step in component creating is producing binary executable code. The relationship between different components is described by the complete waveform. The connection between any two components represents the data or control exchange. We construct the model of application waveform, which is a trivial task. The components created are selected which we want to be used by the waveform and connect them together. To find a component, CORBA namingservice is used. Finally, we export the executable file packages of the application that includes: domain profile and executables. These files are used by radio management in deploying. The AssemblyController is an essential part of all the waveforms. Finally, the waveform is deployed in a group of components in Assembly Descriptor, that describes component interaction. Each component is required to tell about its resource and capacity requirements which is checked by Radio Management. This components deployment task is performed by Radio Management that enforces a life cycle for all the deployed components.

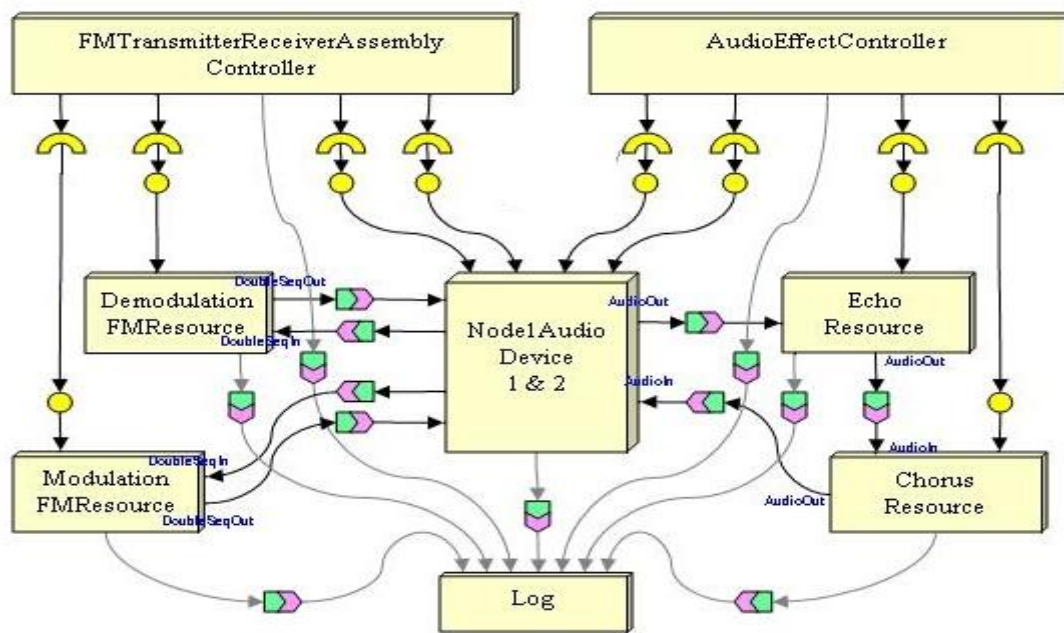


Fig. 5. SCARI example waveform.(taken from [18])

This flow is described in detail in SCARI Waveform Application Builder User's guide that can be downloaded from [18], together with other useful documentation. An example waveform is provided in this documentation that consists of eight components: An AudioDevice, An AudioEchoResource, An AudioChorusResource, An AudioEffectController, A ModulationFMResource, A DemodulationFMResource, A FMTransmitterReceiverAssemblyController and ALogImpl. The assembled waveform is shown in Figure 5.

5.2 OSSIE [7]:

Open Source SCA Implementation::Embedded (OSSIE) [7] is an Object Oriented SCA Operating Environment, developed at Virginia Tech University, USA. It is intended to provide an open source SDR platform and is written in C++ using the omniORB CORBA ORB. This object-oriented SCA OE (Operating Environment) works on a Linux operating system. The software package includes a SDR Core Framework based on the JTRS SCA, tools along with signal processing components and waveforms (applications), device interface software and node configuration file for use with OSSIE. It runs on Intel and AMD based PCs and a release including enhanced support for embedded as well as PC-based applications has also been added. This project uses Xerces XML parser and is currently migrating to tinyXML. The tools autogenerate component source code and supporting files. OSSIE enables an easy transition from concepts to real implementation in SDR design for engineers who may or may not have a strong background in programming. The OSSIE project provides the software infrastructure mandated by the SCA and easy to use tools for application and component development. These tools help with application generation, to reduce the need for manual programming. There is a simple demonstration waveform called “*ossie-demo*” together with several lab exercises. The *ossie-demo* waveform is a simple baseband simulation of a quadrature phase shift keying (QPSK) communications link. It has three components: The TxDemo which is a transmitter that repeatedly sends a test vector of 1024 bits. The signal from TxDemo is then processed in the ChannelDemo component, which adds white Gaussian noise and imposes phase shift on the signal. The RxDemo

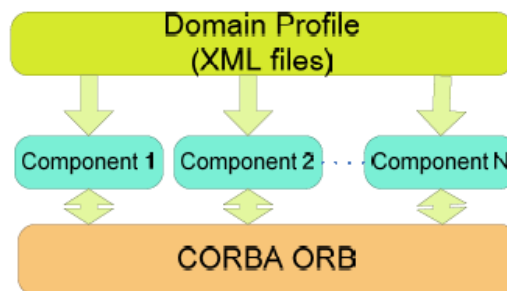


Fig. 6. OSSIE Block Diagram

component demodulates the received signal and compares the demodulated data with known test vector to calculate a bit error rate. The Signal Processing Library of OSSIE implements Digital Signal Processing Algorithms commonly used within radio-communications to create different components. The components for transmitter/receiver are reconfigurable. Figure 6 illustrates the block diagram. The components are the radiocommunication algorithms, whereas Domain Profile comprising of XML files is shown in Figure 4. We explain the functions forming these components in SigProc namespace explained in Chapter 4.

5.2.1. Waveform Workshop – Tools of OSSIE [8]:

The waveform workshop of OSSIE has a comprehensive set of tools meant for rapid development, testing and configuration of SDR waveforms and components. The OSSIE Waveform Developer (OWD) simplifies component development. It is even easier than SCARI to develop components and waveforms in this Reference implementation. OWD provides a graphical user interface allowing the developer to design new software components and interconnect existing components to create waveform applications. OWD uses a template based approach to code generation, which results in XML files. The component source code is automatically generated in C++, in such a way that the application developer does not need to worry about the SCA CF and CORBA. He only needs to add the desired signal processing functionality.

ALF is application visualization and debugging tool that allows a developer to launch the waveform on target platform, display it in block diagram form and probe the components port. The capabilities were further enhanced by adding more plug-in tools and the capability to rapidly launch software components as applications. The interconnection between two applications running simultaneously is also possible by means of a connection tool. This allows adhoc applications to be built from independently running components. The connection tool uses XML files to define producer and consumer applications, together with their installation methods. An improved user interface and other more desired features, such as code editing and debugging, support for collaborative development through interfaces with revision control systems were added by Eclipse. The OSSIE Eclipse Feature (OEF) makes it possible to develop components and waveform applications, launch the domain manager and device managers, launch ALF and if desired the legacy OWD, all from Eclipse GUI. The Waveform Dashboard (WaveDash) exploits SCA functionality to provide an interactively customizable GUI that provides easy control of waveforms, while creating them. The property values of each component's instance can be set to interface specific defaults in waveforms developed using OEF, and can be set interactively at run time using WaveDash. The GUI allows to install and run waveforms, configure the component properties and choose the properties to display and the type of control to be used for each displayed property. This helps in reducing the Software programming errors. In *ossie-demo*, by the help of WaveDash and ALF, the user can interactively vary the standard deviation of the noise and the phase shift imposed by the ChannelDemo component while observing the signal constellation.

6. Commercial SDR [19]:

We have already given two examples of SDR from academic domain. The exploitation of SDR technology in actual commercial products has evolved more slowly than was expected some time ago. Nevertheless, today SDR commercial devices are most oftenly used in rugged and non-rugged military

and aerospace equipment, cell phones and wireless network cards, PCS Base Stations, Signal Intelligence Radar, Sonar Applications and Satellite Communications. Also, soon they are expected to be utilized extensively in user terminals. As an example of commercial available SDR products, we quote the SDR-4000 which is a product of Spectrum Signal Processing by Vecima, a part of Vecima Networks Inc. and is a leading developer of Software Defined Radio solutions for communications and intelligence gathering applications. We give below the product portfolio taken from [11] for the sake of completion as our discussion will not be complete if we don't give any commercial product employing this technology.

6.1 SDR-4000 [19]:

The SDR-4000 is engineered by Spectrum [19]. It is a small form factor transceiver that helps to develop and deploy wireless modem solutions for tactical military communication systems. The SDR-4000 SCA option provides an SCA Board Support Package, SCA v.2.2 Core FrameWork. Spectrum's quicSpin design methodology is supported, enabling rapid optimization of size, weight, power consumption, cost and ruggedization based on specific program requirements. SDR-4000 cards are designed for harsh environments. It is because of this reason that they support conduction-cooling, extended temperature range, and increased shock and vibration immunity. The SDR-4000 comes with a standards-based software environment including Spectrum's quicCommTM (software development library) hardware abstraction layer and software library. The quicComm software abstracts the underlying hardware. This feature provides users with basic transport level access and control of Spectrum's flexComm products. This in return accelerates user application development. The other features include a real-time operating system with an integrated development environment, a CORBA ORB, optional SCA Core Framework, SCA development tools, and for qualified customers, an early support package option for the JTRS MHAL APIs. The SDR-4000 product portfolio consists of a series of 3U cPCI-based carriers and XMC modules, software, development systems defined below:

- **PRO-4600:**

With the help of a high-speed communications fabric, a 3U cPCI SDR processing engine that supports a Xilinx Virtex-4 user FPGA, TI TMS320C6416T DSP, and Freescale MPC8541E General Purpose Processor (GPP) is integrated.

- **XMC-3321:**

The XMC-3321 supports an onboard Xilinx Virtex-4 user FPGA for wideband processing. It is a dual channel trasceiver XMC module that supports IF-to-digital conversion via two 14-bit A/D converters sampling at up to105 MSPS and digital-to-IF conversion via two 14-bit D/A converters upto 300 MSPS in a single-width XMC form factor. This way the industry standards 10.7, 21.4 and 70 MHz IF frequencies are supported . It is optimized to operate in two modes: with the PRO-4600 for Tactical

MILCOM applications or as a stand-alone XMC module on other industry standard XMC-compliant cards. The XMC-3321 has also the capability of operating on Spectrum's ePMC carrier cards by exploiting Spectrum's Solano high speed communication technology.

- **Integrated Development Systems:**

An air-cooled system that integrates the PRO-4600 and the XMC-3321 in a single chassis is provided. A development PC hosts the quiComm software that abstracts the underlying hardware and all of the tools necessary to quickly start the development on the SDR-4000 platform.

- **SCA and ORB:**

This is an optional specification and meant for systems requiring SCA compliance. In that case, an operational SCA core framework and the toolkit associated with it is preloaded on the SDR-4000.

- **The quicSpin Architecture for Rapid Optimization:**

The SDR-4000's unique design supports Spectrum's quicSpin architecture which is based on Spectrum's tactical MILCOM reference designs. The architecture of two cards, PRO-4600 and XMC-3321, uses a modular hardware and software design. This enables quick optimization of size, weight, power, cost and/or ruggedization characteristics to meet the specific requirements of fielded applications.

- **Pro-4600 and XMC-3321 Black-side Processing Subsystem:**

The SDR-4000 Tactical MILCOM black-side processing subsystem supports the modem, link and network layer processing of a SDR. The two cards, PRO-4600 SDR and XMC 3321 operate together to support up to two IF channels simultaneously within a single 3U cPCI slot. Based on Spectrum's MILCOM reference designs, the PRO-4600 and XMC-3321 have been architected to support the black-side digitization and processing of complex waveforms. This includes waveforms that require low latency deterministic operation necessary to maintain synchronization on the frequency-hopped Tactical MILCOM network.

- **Hardware and Software:**

The PRO-4600 is a 3U cPCI heterogeneous processing engine, employing a combination of Xilinx Virtex-4 FPGA, TMS320C6416T DSP and MPC8541E GPP to support the black-side processing requirements of size, weight and power limited SDR applications.

- **Software Operating Environment:**

SDR-4000 features a standards based software operating environment. The code portability is provided by the software stack of the operating environment (OE), by providing a choice with respect to the component usage in an application. The operating systems supported are designed for use in embedded systems that require maximum reliability. The two supported operating systems are the Green Hills INTEGRITY real-time operating system (RTOS) and the Wind River VxWorks RTOS.

- **FPGA Tool Flow:**

The SDR-4000 is designed to support the Xilinx ISE Foundation tool flow. ISE is an integrated programmable logic design environment that includes schematic capture, power analysis tools, physical synthesis for FPGAs, advanced Place and Route Algorithms, and COREgenerator, a graphical interactive design entry tool that is used to create high-level modules. Apart from Xilinx ISE, there is a possibility of usage of other tool sets, as well.

- **Software Communication Architecture:**

The SDR-4000 supports the widely adopted SCA Core Framework from CRC Canada, the SCARI Core Framework that we have already explained in Section 5.1.

The Block Diagram of SDR-4000 is shown in Figure 7.

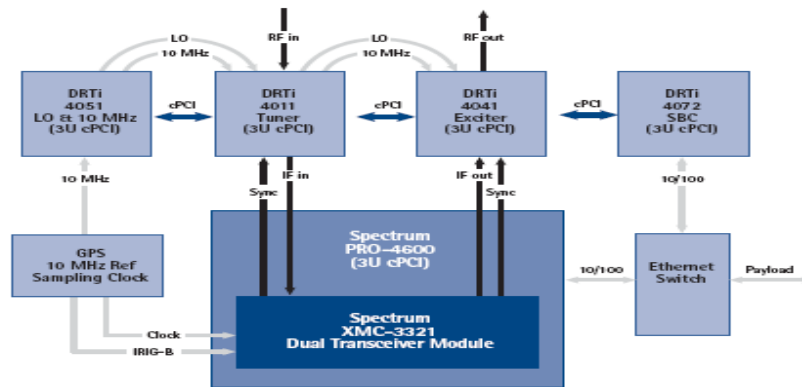


Fig. 7. SDR-400 Block Diagram. (taken from[19])

There are many other commercial SDR based products available, however we chose to study in detail SDR-4000 of Spectrum Signal Processing as an example of commercial product.

The detailed study of the notion, SDR, that we have done in this Chapter shows that this technology has left the experimentation phase and different aspects of this technology are efficiently being addressed by the research community. We gave an overview of these aspects, together with the key-feature definitions defining this concept. We carried out a detailed study of SDR especially emphasizing the recent times effort to address the implementation issues.

7. Conclusion:

Starting from the notion of ideal Software Radio and its degradation to Software Defined Radio, this chapter threw light on the important features and research issues in SDR, which clearly show that this technology has many positive signs, especially the SDR based projects and products entering the academia and commercial market. Although, the technology has evolved at a slower pace than

expected some years ago [20], it has clearly left the theoretical and experimentation phase started by military and communication systems and is now, fielded in hand held devices. A fundamental challenge for SDR products is that of providing the essential features that we discussed in this Chapter. A modern SDR Architecture has been described, based on the concept of Radio Computer that supports modular design. The US military's SDR, JTRS based on SCA and CF concept is explained. Some SDR examples in both, academic and commercial market are studied. As academic SDR, CRC's SCARI and Virginia Tech's OSSIE have been studied deeply. The Commercial SDR studied was that of Spectrum Signal's SDR-4000.

We saw in this Chapter that the ability to reprogram a system facilitates hardware reuse even when a new generation of hardware platforms is available. This phenomenon is likely to be experienced by mobile terminals in near future. Chapter 3 explains the concept of Cognitive Radio and different machine learning techniques applied to realize this technology, together with an overview of relevant projects.

Chapter 3:

Cognitive Radio - State of the art:

1. Introduction:

Having defined SDR in Chapter 2, we proceed in this Chapter to understand the notion of Cognitive Radio (CR) that adds intelligence to SDR, by cognition incorporation. Before proceeding any further, we would like to define the terms intelligence and cognition, as per the literature [22, 23, 181].

- *Intelligence* is the capacity to think rationally and act purposefully in order to deal effectively with the environment. This capacity can be natural (pertaining to an individual) or artificial (pertaining to an agent or machine).
- The intelligent process of gaining comprehension by awareness, perception, reasoning and judgment is called *Cognition* and the machine employing this gained comprehension for problem solving is referred to as *Cognitive*.

CR is linked with the idea of SDR as it can be perceived as a kind of SDR with some *intelligent* functionality. This intelligent functionality is meant to sense the changes in environment and then react smartly according to the changes, apart from predicting the future expected trend based on “*Past predicts the Future*”. There are many factors that demonstrate the need for CRs. For example, the Spectrum has proved to be extremely precious resource in today’s resource competitive environment. For hundreds of years, the Spectrum was present everywhere and was abundant irrespective of place or time, on the earth. There was no need of license and neither existed any license granting authority. The exploitation of Spectrum was based on *static principles*, i.e. a certain portion of the radio spectrum is licensed to a specific party in a specific region on long term basis. This static approach provides interference free solution for the licensed party. However, the sporadic use of the precious resource of Spectrum leads to gross inefficiencies, forcing us to reconsider the principles based on static allocation. The sporadic use of Spectrum can be witnessed in the spectrum from 6 KHz to 300 GHz, which is fully allocated, at any given time, but most of it remains un-utilized. Furthermore, new wireless systems: radio, television, mobile phone, wireless internet connectivity have strong dependence on ever improving and dynamic technologies. This demands communication devices which are able to communicate with heterogeneous systems. This is where the concept of dynamic

spectrum allocation comes into play. This dynamic allocation is one of the most important functionalities of CR. This will enable to increase the total system capacity by looking into different techniques of using CR to exploit the local unused spectrum. The trend in this evolution towards the new era of communication is undoubtedly the novel approach to radio resource management, based on Dynamic Spectrum Access (DSA) or opportunistic access to underutilized and free frequency bands. As a part of this evolution and the growing convergence of wireless communication systems, it is necessary to review different concepts and architectures of radio transceivers and their components, taking into account the key factors such as reconfigurability and adaptation to space-time spectrum availability and the actual user's transmission request. These are the main attributes of the concept called "Cognitive Radio" (CR). This technology holds the promise to solve the radio spectrum scarcity and aims to provide heterogeneous connections in future wireless communication networks. We address our efforts in this DSA scenario in Chapter 6, using Artificial Neural Networks.

Another aspect is the Multi-standard Transceiver Design. The modern commercial multi-standard mobile devices employ multiple transceivers, where each one is dedicated to an individual communication standard, e.g. Global System for Mobile Communication (GSM), Universal Mobile Telecommunication System (UMTS), Bluetooth, Wi-Fi 802.11 a/b/g/n, Worldwide Interoperability for Microwave Access (WiMAX) 802.16e, Long term Evolution (LTE) and other wireless systems. The approach of separate transceiver engineering for individual modes of operation is straight-forward and it provides the best performance for each node, but on the other hand, this approach significantly penalizes the overall circuit complexity and hence the power consumption and implementation costs are increased. To accommodate CR technology, a versatile multi-radio transceiver, capable of generating and processing any required waveform according to the given conditions, becomes the ultimate goal. These requirements for a Multi-Standard CR Transceiver call for a high degree of reconfigurability and adaptability at each and every stage of Cognitive Radio design. At the same time, the requirements of low power and implementation costs should be assured. We address our contribution in this aspect of Transceiver Design in Chapter 5, using Artificial Neural Network Techniques. There are many challenging research topics that are currently dealing with the cognition incorporation in the radios, notably design of a Multi-standard Universal Transceiver that takes into account all features of all the layers and adding efficient Spectrum management techniques. The CR portion of this thesis addresses both of these issues by proposing different Artificial Neural Networks based solutions.

The cognition cycle state diagram modified by Haykin [23] is shown in Figure 8 that consists of three states. The cognition cycle starts with the passive sensing of RF stimuli and terminates with action. The three states shown in Figure 8 are:

1. Radio-scene analysis, which encompasses the estimation of interference temperature of the radio environment and detection of spectrum holes.

2. Channel Identification which encompasses the estimation of channel-state information and prediction of channel capacity for use by the transmitter.
3. Transmit-power control and dynamic spectrum management.

Dynamic Spectrum Management is not the only functionality of CR. A radio in which the environmental awareness is limited to only dynamic spectrum management can be seen as a subset of cognitive radio. This opportunistic radio [24] limits the environmental awareness to spectrum management sensors. In fact cognition cycle of Figure 8 is a modified version by Haykin, stressing the spectrum sensing. The original cognition cycle diagram, introduced by Mitola [22,25,30] talks of an environment aware general sensor, autonomous for decisions regarding optimized interfaces to surrounding systems. This original cognition cycle has a much wider definition and functionality (including spectrum sensing), as compared to the modified cognition cycle whose main focus remains spectrum related sensors. We will delve into other sensor details in Section 1.1 after giving the formal definition of Cognitive Radio. Nevertheless, FCC has recommended that significantly greater spectral efficiency could be realized by deploying wireless devices that can coexist with the primary users. In this respect, equipping the device with CR capabilities is meant to achieve efficient spectrum management.

Figure 8, the state transition diagram of CR, is meant to show that CR based systems behave and predict the future in a proactive manner, i.e. based on external stimuli and previous observations, apart from being governed by their goals, principles, capabilities, experience and knowledge. Haykin's cognition cycle presented in Figure 8 is meant to describe this very cognition behavior.

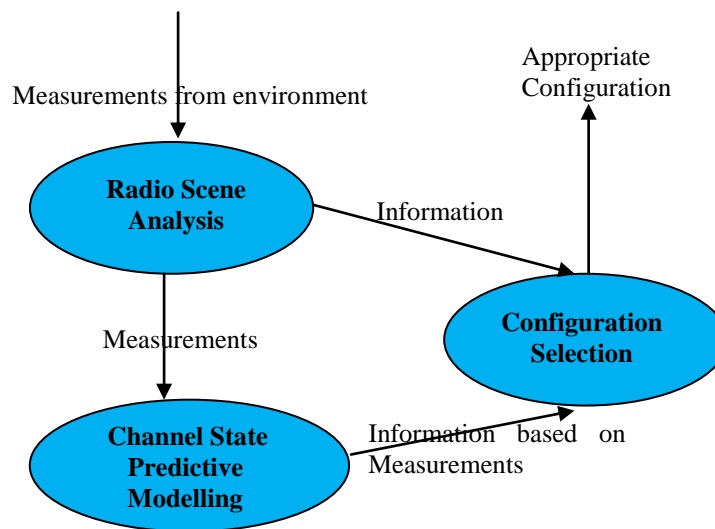


Fig. 8. Cognition Cycle [23]

1.1 Definition:

CR was first described in [22, 25, 30] as *a decision making layer covering many sensors across all the seven layers*. Mitola was not only the creator of SDR notion, but the terminology of Cognitive Radio was also coined by him. Another important definition was given by Haykin [23] as: *A CR is defined as inclusive of SDR to promote efficient use of spectrum by exploiting the existence of spectrum holes*. The cognition cycle state diagram is based on this definition. This definition restricts the cognition in the radio to a single sensing functionality: Spectrum. However, as Mitola defined, Spectrum sensing is only one of the functionalities of CR. The definition mostly quoted by the researchers is that of Wireless Innovation Forum. In this thesis, we stick with this definition, which says:

A radio that can sense, be aware of and learn from its environment in order to adapt its operating parameters accordingly, leading to optimize radio resource use. [1]

This definition tells that future Cognitive Radio devices will have the capability to optimize their state, thereby adjusting their operating parameters accordingly, after sensing the environment variables. Thus the discrepancy or paradox between spectrum allocation and spectrum use could be overcome by allowing more flexible usage of spectrum. However, we would like to distinguish between CR and DSA as the scope of CR is much broader than just exploiting the spectrum holes. The cognitive functionality may be spread across the layers of the communication architecture. This results in coordination amongst the layers for a more efficient use of radio resources. The power spectral density is a measure of average power distribution as a function of frequency, therefore it is used as a sensor to determine the channel bandwidth. The sensor to distinguish between direct sequence and frequency hopping as addressed in [129] uses time frequency analysis. In the same way, Guard Interval detection may be used as a sensor to distinguish between single and multicarrier systems as done by Hachemani in [126]. We have already given the example of spectrum sensor in the beginning of this Chapter. We will explain more about the sensors used to sense the vacant or occupied band, in Chapter 6. Cyclostationary feature sensing is another sensor that tells about the considered frequency band status. The related sensors to cyclostationary process are cyclic frequency, carrier and symbol frequency and positioning and localization characteristics (in the frequency domain, for example) that can be used to sense different aspects of the received signal. Another important physical layer sensor talks of modulation type recognition to sense the RAT. The other sensors, related to physical layer, known to manage the system behavior intelligently, talk of coding methodologies, handover mechanisms and channel estimation. Likewise, QoS can serve as an application layer sensor by considering speed, video quality, price and other such user oriented personal parameters. Thus, even in today's spectrum constraint environment, dynamic spectrum access and sensing occupies a vital importance in CR Engineering, the cognition cycle continues to envision cognitive radio which is much more than just

spectrum sensing. One of the key factors that we need to underline is that the CR behaves according to its previous observations, or we can say that it learns some lessons with the passage of time, that it applies in the future. In this respect, a general scheme of the CR reconfiguration cycle is shown in Figure 9 that emphasizes on its learning character. First of all, the actual transmission request is evaluated and spectrum sensing and signal waveform determination functions are employed in order to find proper transmission resources (unused or underutilized radio resources referred to as *spectrum holes* or *white spaces*). This function may be based on real-time spectrum scanning by individual CR terminals or by CR nodes, and moreover, the information on spectrum availability in a given region can be obtained from local databases (e.g. a regional frequency allocation database and actual utilization statistics downloaded directly from a CR node). Next, the best course of action upon spectrum findings and spectrum availability is taken. This action includes a new allocation of resources and process initiation. Finally, the *reconfigurable cognitive transceiver* is adjusted accordingly. This whole process is dynamic and has to be updated regularly, according to the space-time spectrum availability and the actual transmission request such as voice vs. data service, data rate, Bit Error Rate (BER), latency, etc. The Chapter 5 is dedicated to the notion of reconfigurable cognitive transceiver. Here, we need to make a point that the regular updating, discussed above is based on the lessons learnt from the past. Artificial Neural Networks are the techniques that employ the learning mechanism to solve different problems.

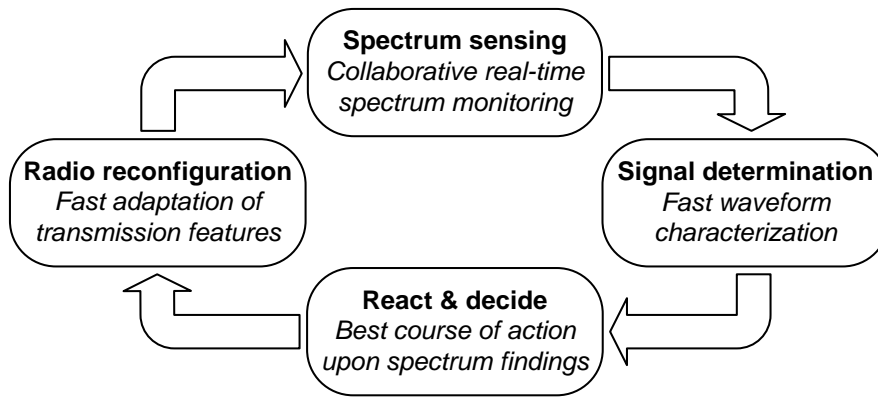


Fig. 9. Reconfigurable Cognitive Radio Cycle.

2. Theoretical Research Issues:

There are numerous research issues as CRs pose challenge at all levels of abstraction and has proved to be an extremely interdisciplinary topic. It is because of this reason that it is difficult, rather not possible, to provide an exhaustive analysis of all research works available on Cognitive Radio communications. The purpose of this Section is, therefore, to realize the obstacles in the way of recognizing the communication environment and independently adapting the parameters accordingly

to maximize the quality of service (QoS) in such a way that optimizes the different radio resources: spectrum, battery, carrier frequency usage. The issues discussed in this section are open and currently a question of research. These issues have a direct impact on the feasibility of CR.

2.1 Sensing:

The ability of an opportunistic system to sense the existence of other systems, be it another opportunistic system or the priority user, is a vital issue in CR domain. It is one of the defining functions and the most important issue as the ability to sense the radio channel in order to find opportunities in spectrum and adapt the radio parameters is perhaps the most important task of CR. Recent measurements have shown that the spectrum usage is concentrated on certain portions of the spectrum while a significant amount of the spectrum remains unused. Spectrum sensing has been identified as a key, enabling CR to not interfere with primary users, by reliably detecting primary user signals. Therefore, sensing requirements are based on primary user modulation type, frequency and temporal parameters. Spectrum sensing is often considered as a detection problem. The performance of any sensing algorithm can be characterized by variables, such as detection probability and false alarm probability. An improved detection probability can lead to a higher protection level to primary users, while a lower false alarm probability offers better opportunistic access to secondary cognitive nodes. Many sensing techniques are available in order to detect the holes in spectrum band, e.g. Energy Detection, Matched Filter Detection, Waveform based Sensing and Cyclostationary Feature Detection. We will further explore these techniques and exploit Cyclostationary Feature Detection to propose a neural network based algorithm to sense the spectrum beforehand, later in this thesis. Cyclostationary models have been shown in recent years to offer many advantages over stationary models. Cognitive Radios should decide on the best spectrum band to meet the Quality of Service requirements over all available spectrum bands, therefore spectrum management functions, followed by Sensing techniques are required for Cognitive Radios. We deal with this Spectrum management problem by predicting the future spectrum occupancy status based on previous observations. The significantly greater spectral efficiency is realized by Sensing and we address this problem in detail in Chapter 6.

2.2 Decision making process:

Cognitive Radio has to *decide* to operate in a specific configuration. The decision has to be based on appropriate *reasoning*. It is because of this reason that a reasoning algorithm (also called as Reasoning Engine) that applies the knowledge to the current state of the system and reaches one or more conclusions is a necessary part of all the Cognitive Radio Architectures. The decision process should be based upon a reliable and smart reasoning algorithm. The algorithm should be capable of modifying the system operation based on applications of knowledge to the combined state. The reasoning should be efficient enough so that the existing set of knowledge is applied to a current situation to result in the

identification of a course action. At the same time, this experience should be used together with the previous knowledge to predict the future expected situation and prepare in advance to act. This *intelligent* behavior is what makes a radio *cognitive*. A CR should intelligently adapt its operational behavior by acting efficiently to external and internal stimuli. However, such a system has a critical limiting factor: the system can adapt based upon solely predefined behaviors, stored as previous applied inputs in the knowledge database. Thus, even though the radio responds intelligently to external stimuli or change in environment and internal state changes, it can adapt its behavior only within the limits of previously defined knowledge. Thus, it cannot adapt to new and unexpected situations. The next issue *learning* deals with such a behavior.

2.3 Learning Process:

In order to adapt to new and unexpected situations, as in previous Sub-section, a learning process is required. Learning is a process of perception, observations and actions. The Haykin's cognition cycle is a constant learning mechanism, during which it adjusts its operating parameters, observes the results and eventually takes actions, that is to say, decides to operate in a specific radio configuration, expecting to move the radio towards some optimized operational state. The learning mechanisms used to train the cognition cycle should be capable of exploiting measurements sensed from the environment, gathered experiences and stored knowledge. We have used Artificial Neural Networks with different learning mechanisms (Back-propagation, Levenberg-Marquardt, Self-Organization) which can be roughly classified in two categories: (1) Supervised Learning (2) Un-supervised Learning. Depending on the information availability, different learning solutions can be applied. Apart from ANN based learning algorithms, other machine learning techniques also possess some kind of learning phenomenon, e.g. the existing learning in games theory literature provides a broad spectrum of analytical and practical results on learning algorithms and under-lying game structures for a variety of competitive interaction scenarios. However, when selecting any learning solution for wireless networks, the specific constraints and features of the network and the radio are needed to be considered. As an example, the learning algorithm that should be deployed by a user in a wireless environment strongly depends on what information a secondary user can observe about the other secondary users, given the spectrum regulation rules. We will explain the learning mechanisms using Artificial Neural Networks in detail for CR Transceiver Engineering and Spectrum Prediction in Chapters 5 and 6, respectively. It must be noted that each adapted learning mechanism suits the specific requirement that we will explore.

2.4 Architecture Implementation:

While the OODA (Observe-Orient-Decide-Act) Model proposed by Mitola has centered on Decision making and learning issues, within a cognitive architecture, it has proved to be a reactive architecture and not a proactive one. This means that it is designed to wait until an event occurs, before changing

the configuration. To encounter this, there exists a proactive architecture proposal CECA (Critique-Explore-Compare-Act) [26] that studies the broader aspect of making changes before they are needed. CECA is an expansion of OODA. However, there have not been many efforts in this area especially in exploiting the prediction algorithms used in Artificial Intelligence domain. The biological models such as ant colony optimization, particle swarm optimization, artificial neural networks together with the mathematical models such as hidden markov models, branch prediction from computer science and grey system theory prediction algorithms have required potential to enhance the reactive CR architecture into a proactive one. There exists other architectural proposals as well, making it more difficult to decide for a specific architecture that supports CR and related design.

2.5 Equipment Test Procedures and Certifications:

In Sub-section 2.6, of Chapter 2, we discussed the certification issues for SDR. In CR paradigm, the radio with cognition capabilities, poses difficulties with respect to standardization and certification procedures. The operational limits are difficult to be estimated and in a way contrary to the concept of flexibility required in a CR design. The testing procedures to detect the interference of a cognitive device with primary user of a given frequency slot should be known to the vendor. Also while SDR platforms are not a strict requirement for creating a CR, the architecture flexibility of a SDR is one motivation for pursuing the SDR based implementations of CR. The standardization and certification process should compare and analyze different CR architecture implementations and clearly test for the incorporated intelligence quantification, in terms of making the device proactive above a given threshold.

2.6 Interaction with all Layers of Protocol Stack:

Different layer parameters need to be considered to optimize the cognitive decision, e.g. application layer parameters such as localization and positioning characteristics, a part from physical layer and transport layer features of a signal. Each layer has its own design criteria and different constraints. Therefore, formalizing the different objectives from different layers together and putting them onto a common mathematical platform is a difficult task.

2.7 RF Design:

A CR design, by definition is different as the RF frequency and the bandwidth is not known in advance. This is in contrast to a conventional radio system. In a traditional radio system, we assume about the interferers keeping in mind the worst interference possible and keeping this consideration in mind, the design specifications with respect to RF are determined. To this effect, the traditional radio systems use a pre-select filter at the receiver side to limit the interferers. As we said that, in a CR we have no estimation about these parameters, making the traditional approach less pragmatic, which exacerbates the RF design issues. The RF designers are meant to make the CR as flexible as possible

with respect to radio frequency choice. Thus the design is susceptible to interference as pre-select filters are required to be removed. Also the fact that all RF specifications cannot be mapped on the circuit blocks without the information on interferer scenarios is another problem for RF design engineers. There exist methods of receiver design and frequency planning that lead to excessive circuit block requirements, knowing that there is no pre-select filter at the receiver side. However, it results in an increase of the requirements of circuit block. The defining cognition capabilities of a CR should be exploited to deal with these strict design requirements. The interference related information in a CR should be used to select the RF frequency, which should not be limited to the spectrum occupancy but also on the suitability of a given frequency for communication. It will eventually help in avoiding the circuit block requirements to become over demanding and at the same time will not limit the capabilities of a CR.

2.8 SoC Implementation:

The System on Chip design is an important research issue as it constitutes the question of searching for a compromising spot between hardware and software processing. The memory needs for different functional units are high as it is a frequency agile system. A deep understanding of different algorithms implemented in software is needed. There are many options for a specific algorithm and hence a huge design space. The next aspect is integrating different algorithms together to work as a waveform. This is a challenging task as the integration has to respond the real time constraints. High level of flexibility and high processing power is needed, when the power consumption is a critical issue. The integration phase has some other important principles, e.g. the I/O bandwidth is a significant issue. The distinction between system level and pure processing level should be explicitly made. The software programming of such an architecture where the compromising distinction between hardware and software has to be made is a non-trivial task, which requires several dedicated tools. This reconfigurable computing is important because it allows the computational capacity of the machine to be highly customized to application needs and to be reused in time. In [27], such a reconfigurable heterogeneous hardware and its reconfigurable architecture is discussed, in which the SoC contains a general purpose processor, FPGA and a coarse-grained reconfigurable part. The implemented SoC is claimed to yield a combination of performance, flexibility and energy efficiency.

2.9 Accurate and Secure Primary User Detection:

CR exploits the un-utilized bands after sensing the spectrum. The secondary user is permitted to operate in a licensed band, provided that it does not cause any interference. Apart from that, the secondary user is required to watch for the existence of primary user signals in the operating band or future expected band. In case, when a secondary user detects the primary user presence in the band, it must vacate the band immediately or switch to another available opportunity. At the same time, if the secondary user detects the presence of another opportunity seeker, it should invoke a co-existence

mechanism to share spectrum resource. In this respect, primary user emulation attack is defined in [28] as an attack in which a malicious secondary user attempts to gain priority over other prioritized users by transmitting signals that emulate the characteristics of a primary user. There should be some way to avoid this emulation attack. Energy detection is one of the simplest methods to secure primary user detection. Other spectrum sensing approaches, feature detection exploiting cyclostationary features or matched filter detection are used to detect the specific properties of primary user. The accurate and secure primary user detection is a challenging problem and is defined as the most important aspect of a CR by which it is made necessary for secondary user to vacate the opportunity as soon as it detects the primary user.

2.10 Interoperability:

Interoperability, in CR scenario, refers to sharing of radio resources between multiple operators. This sharing may be opportunistic or managed. The basic research focus for interoperability issues is to maximize the radio resource usage with co-existence and co-operation with each other. One of the major radio resource sharing, i.e. Spectrum is the major attribute of CR networks. A CR should incorporate interoperability so that it may be capable of automatically configuring itself to communicate between incompatible radios. The detection of presence of spectrum opportunity in the vicinity, followed by opening up lines of communication in real time constraints between two different radio groups is the basic notion behind the interoperability issues. Different machine learning techniques can be applied to achieve this interoperability between two incompatible radios.

2.11 Security:

The misuse of CRs should be avoided at all costs by incorporating some robust security measurements. The CR software has unique properties that make it different from conventional software. These special properties make the CR more vulnerable to attacks. These attacks may cause harmful effects, for example transmitting at higher power than allowed, launching attacks against primary user networks or unauthorized changes to CR operating parameters such as power, frequency and modulation. At the same time, the rigid real time requirements of radio systems prohibit the use of encryption techniques. Guarantee of secure and functionally correct embedded software is critical to the success of its deployment on CR. Even a slight error in the software can make the radio design vulnerable to attacks that may crash the system or produce misleadingly erroneous results, such as unwanted transmission in bands that are critical for defense applications. It is, therefore, important to ensure the security and reliability of CR by identifying and fixing the vulnerabilities by secure protocol design for different cognitive functions, such as spectrum access by secondary user. Despite the fact that several researchers have begun working on security implications [29], this research area is still in its initial phases.

In this Section, we described the research issues that need to be addressed to make the CR technology 100% viable. We can divide the research issues addressed in the five main categories: Computation-related problems, Architecture related problems, Implementation related problems, Physical layer related problems and Protocol related problems [182]. Each problem's sub-classes are formed and described in detail. We have seen that CR Engineering is a multi-disciplinary topic and hence to address the problems, one needs to develop a vast knowledge about different aspects of multiple disciplines. We will delve into further details about this in the future Chapters. (Chapters 5 and 6).

3. Cognitive Radio Architecture:

The cognition cycle presented by Mitola and modified by Haykin, identifies the processing structures for the integration of sensing and perception into radio. The CR Architecture articulates the functions, components and design rules of next-generation cognitive radios. Thus, CR Architecture can be perceived as a framework that helps the evolving components to integrate into an ever evolving sequence of designs specified by engineer within specified constraints [31]. A powerful cognitive architecture should facilitate a product which has three features: rapid, cost effectiveness and service evolution. In that respect, the radio architecture can be classified in two categories: an *open architecture* that is available to the public, while a *proprietary architecture* which is the private intellectual property of an organization, government entity, or non-public consortium. Mitola's proposed OODA Architecture (Sub-section 2.4 of this Chapter) is an open architecture that incorporates the spectrum and network sensing. An extension to this OODA Architecture is The Virginia Tech's Case Based Reasoning Cognitive Engine [26], which is developed for 802.22 Standard where secondary users are obliged to quit the band for a primary user. A Multi-objective optimizer takes care of flexibility requirements between different aspects of a CR. OODA Architecture is further modified into CECA to answer the today's commercial evolution in RF chip sets. Today's commercial RF channel sets have typically four chip sets (GSM 900, GSM 1800, CDMA and Blue Tooth), and it is expected that in near future, it will evolve to a dozen band-mode combinations with ever smarter MIMO emerging antennas[32]. Thus CECA Architecture proposes to make the configuration changes even before they are required. Also, a channel set may include a cable interface to the public switched telephone network as well as a radio access point. It is possible that any function may be null in any realization. This eliminates the related components and interfaces from a given product, for product tailoring and incremental evolution. The Moore's law is continuously progressing. This is resulting in increasing large fractions of functionality which are synthesized in chipsets with software defined parameters; in the FPGAs; in DSPs and increasingly on single-chip arrays of general purpose processors like IMEC Belgium's SIMD4 [33].

Today's CRs and SDR based CRs are usually engineered from reusable code bases of millions of

lines of code, the deployment, management, and maintenance of which poses very serious configuration challenges. In SDR Architecture based CR, the SDR software typically is organized as radio applications objects layered upon standard infrastructure software objects for distributed processing such as the Wireless Innovation Forum's Software Communications Architecture, explained in previous Chapter, which originally was based primarily on CORBA. The Object Management Group's evolved SCA has a platform independent model with platform specific models for software-based communications.

As illustrated in Figure 8, Haykin's Dynamic Spectrum Allocation Model analyzes the radio scene to perform three architectural related tasks: to avoid interference, to look for spectrum holes and to provide channel state information to enhance the transmission. This analysis emphasizes on the need of a CR architecture that is well-aware of the occupants of radio environment. The protocol stack implies that the integration of cognitive nodes into cognitive networks via the universal control channel for centralized control is supplemented by group control channels. This means that cognitive functionality has to be implemented in individual nodes as well apart from having centralized control architecture. Spectrum sensing feature is emerging in this architecture as the key feature that enables the greater frequency agility in the spectrum usage for the best possible QoS. Such an architecture should quantify channel occupancy and recognize opportunities which can be helpful in choosing RF chip set, signal selection in space transmission control and other high performance spectrum management characteristics of the physical layer, for example MIMO operation. CR architecture provides evolving frameworks for research, development and product deployment. There are many challenges and issues in cognitive radio architecture and its engineering, for example, many spectrum measurements reported in the literature do not fully accept or support the feasibility of spectrum sharing. Measurements mentioned in [31] show that 5% occupied spectrum usually do not consider the navigation aids and GPS as it is not detectable via spectrum scanning, but through cross-correlation receivers. Other measurements do not account for the duty cycles of the radar bands where pulsed radar listens for most of its duty cycle, contributing 0.1% to spectrum occupancy but 100 % to airport surveillance. It is because of this reason that Pulsed radar spectrum cannot be shared in an exploitable way. Also usually radar bands are included in the spectrum scan statistics without clear caveats. High gain receiving antennas required by space communications are some 60 feet. Another fact is that the signals from the spacecraft are not detected in the spectrum scans either. The adhoc networking in the seemingly unoccupied downlink band can be disastrous for space applications. The examples of underestimation of spectrum occupancy criteria can be witnessed in the spectrum sensing efforts of Tuttlebee's Virtual Center of Excellence (VCE) conducted by Beach's group at the University of Bristol , UK [34]. The unmethodical spectrum sensing results of the mentioned effort was also confirmed by the data sets of the Crawdad site. Analog AM voice is audible 6 dB below the 0 dB tangential noise floor because of the sinusoidal nature of voice. It is an established fact that radio

propagation is extremely ragged, even in the easier spectrum range starting from 300 MHz until 3 GHz. The reasons for this ragged radio propagation are multipath, knife edge diffraction, fresnel zones, and other such well known occurrences due to well established facts. The potential contributions of the researchers pursuing cognitive radios architecture with respect to space-time constraints of the radio spectrum may not be fully realized until the Cognitive Radio Architectures include high-performance spatial knowledge. At present, there is no technical architecture deployed for real time small space time RF spectrum access CR.

4. Machine Learning:

The research efforts in machine learning techniques have grown tremendously in the past decade, with a significant amount of progress along several research paths. Before choosing the Artificial Neural Networks based solution for CR design, we carried out a feasibility study of many machine learning techniques available in literature. There are multiple strategies that exist to implement any machine learning technique for CR modelling, each with some degree of benefits as well as drawbacks in the context of a cognitive radio system. This section addresses common learning approaches and provides some analysis of their applicability to cognitive radio systems.

There are two types of learning: Supervised and Unsupervised. The most important aspect of any learning mechanism is whether the learning required is supervised or unsupervised. In a supervised learning system, as the name suggests, learning is performed through a set of pre-determined conditions, and the system (Cognitive Radio/Cognitive Engine in our case) is trained to come up with the correct output that matches with the expected output. To this effect, the learning mechanism learns to associate a particular set of input stimuli with a learned response or output. Thus with a similar input, the system knows how to react and what to do. In the case of a cognitive radio system, any of the two techniques may be opted for, as the radio's method of learning. The sole purpose of the learning mechanism is to make the CR remember lessons learnt in the past and act quickly in the future. The key words that we have used to define a CR in the first section of this Chapter are: awareness, perception, reasoning and judgment. However, nowhere the keyword of learning is used. In supervised learning, the desired output is known and the learning mechanism forces the technique used to converge to that known output. Therefore, supervised learning may take the form of a dialog between the user and the radio, in which the radio may develop some new assertion or operational behavior model based on the learning mechanism within the radio and then ask for confirmation from the operator if its concluded results are correct. We will delve into more details about supervised and unsupervised learning using Artificial Neural Networks in Chapter 5.

Conversely, the radio system may extend its knowledge through the learning algorithm and simply add the new knowledge to its existing base of knowledge assertions and behaviors. In operational radio

systems, there is some level of protection that limits the degree of behavioral modifications that a cognitive radio may incorporate without external verification and validation. Till date, generic learning-based CR is an area in which not much principles and rules have been applied. There are various proposals that have used techniques as genetic algorithms, fuzzy logic, markov-based channel prediction algorithm and swarm intelligence to optimize the radio performance by optimizing the parameters. The basic notion of using these techniques remains the same: capacity maximization and dynamic spectrum access. The reasoning engine in CR is modeled using these techniques. We must note that the reasoning engine is known as an expert system in the Artificial Intelligence domain [35].

At any time instance, the cognitive engine/expert system gives the conclusions that are based upon some information defined in the knowledge base, from the previous experiences. These conclusions are extrapolated and are based on “learning”. As these lessons are learnt, the learning engine stores them in the knowledge base for future reference by the cognitive engine. In this section, the learning mechanisms, such as reinforcement and temporal difference are discussed. We delve into learning mechanisms for cognitive radio using Artificial Neural Networks in Chapters 5 and 6. In this section, we describe the important machine learning techniques that are candidates for CR design.

4.1 Memorization:

One of the most fundamental and basic learning mechanisms is memorization. This approach captures a sequence of steps or a response to a specific set of conditions and then, when the same task is again encountered, the previously memorized responses are applied. This can only be an effective method of learning if the range of situations encountered by the radio system is limited and well defined. An example of memorization, in CR context, would be the selection of a particular waveform or specific adjustments to the operating parameters of a waveform in response to measured interference. Although the memorization would allow for multiple response sequence, each associated with a specific interface value, each sequence of actions would be explicitly tied to the measured values. Memorization does not allow for generalization of responses based on similar responses to different measured values.

4.2 Swarm Intelligence:

Swarm Intelligence (SI) is a biologically inspired metaheuristic that deals with the study of collective behavior in self organized systems. To this effect, a cognition system - Observe-Orient-Decide-Act- is treated as a self organized system, making this metaheuristic a very promising candidate for *intelligent* wireless communication systems. Swarm Intelligence is defined as: *The emergent collection intelligence of groups of simple agents* [116]. Although this technique has disadvantage that there is no centralized control structure dictating the behavior of individual agents, the local interactions between such agents often lead to the emergence of a globally accepted behavior of system. Ant colonies, bird flocking and fish schooling are the natural examples of this technique. This heuristic, despite its

drawback mentioned above is useful in cognitive wireless communication as it has the required properties of learning to some extent. As an example, in the popular Ant path finding example, the ants finally are able to select the shortest path to reach the food, without any external control and help and only remembering their previous experiences.

4.3 Genetic Algorithms:

Genetic Algorithms (GA) [36] work on the principle of exploiting the best solution and exploring the search space in multiple directions simultaneously. This feature makes them a viable choice as compared to the other counterpart optimization techniques, such as Hill Climbing and Simulated Annealing. GA uses a vector that represents a given condition. Depending on the success of action taken, the vector is modified using three operators, defined below. However, the vector may undergo alteration, or genetic mutation, in a random fashion. Even though this may introduce potentially large vector sets, GAs may introduce new solutions through the random mutation process. This enables a system to generate new knowledge and evaluate its effectiveness by using empirical data collected through the operational environment. Implementation of GAs has continued to evolve to include the use of a vector of multivalued data items. In effect, each item in the sequence is represented as a discrete variable that can be modified as part of the process. GAs work on the principle of exploiting the best solution and exploring the search space in multiple directions simultaneously. The three operators that it uses are defined as under:

- **Selection:**

This operator selects chromosomes in the population for reproduction. The fitter the chromosome, the more times it is likely to be selected to reproduce.

- **Cross-over:**

It combines features of two parent chromosomes. This operator randomly chooses a locus and exchanges the subsequences before and after that locus between two chromosomes to create two offsprings. For example, the strings 10000100 and 11111111 could be crossed over after the third locus in each to produce the two offspring 10011111 and 11100100. The crossover operator roughly mimics biological recombination between two single chromosome organisms.

- **Mutation:**

This operator randomly flips some of the bits in a chromosome. For example, the string 00000100 might be mutated in its second position to yield 01000100. Mutation can occur at each bit position in a string with some probability, usually very small (e.g. 0.001).

All these operations are being performed on Fitness Function defined as the objective function to be

optimized which provides the mechanism for evaluating each string (optimization can be meant to maximize or minimize). Chromosomes are the main carriers of heredity information and are made of units called genes and each gene controls the inheritance of one or several characters. Each chromosome represents a potential solution and different chromosomes, each representing a solution are collectively termed as Population. The three operators defined above are shown in a design of a Genetic Algorithm used to allocate power and bits simultaneously in a MB-OFDM system [36].

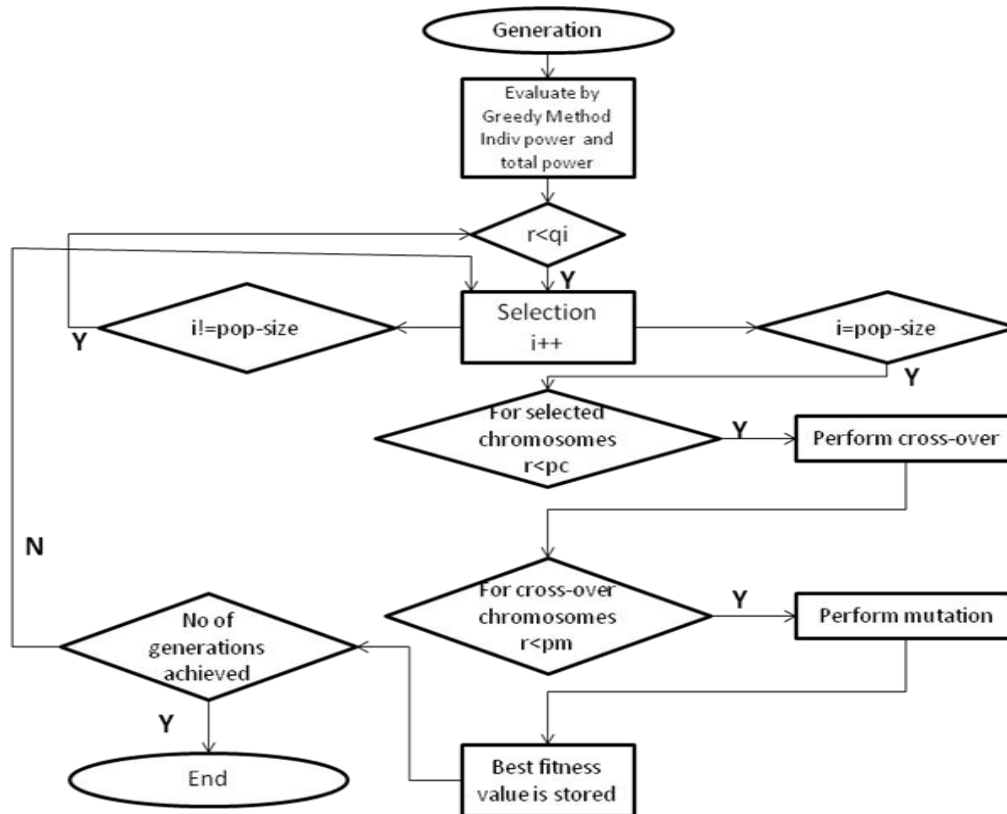


Fig. 10. Genetic Algorithm Structure [36].

4.4 Fuzzy Logic:

Fuzzy Logic based architectures are modular in nature (one of the key aspects of CR Architecture). It is because of this reason that Fuzzy Logic has also emerged as a candidate for CR Architecture design. Fuzzy Set Theory differs with Traditional Set Theory in that partial membership is allowed to all the members for all the sets. This leads to fuzzy logic interface where predicates are treated as partially true/false. Hence a Fuzzy Controller based knowledge database is established which is actually the process of translating the rigid [0,1] measurements into their fuzzy representation. The degree of partial truth is based on fuzzification which means that for each input variable at every control cycle the degree of truth of the input is verified. To this effect, fuzzy controllers are claimed to be the natural

choice for CR modular cross design.

4.5 Other Methods:

There are several other methods with their own sets of advantages and drawbacks which have been reported in the literature to address the different design requirements of CR. Each method is better suited for a given category of problems. At times, the combination of multiple approaches within the same algorithm yield better or faster results. The user requirements and most importantly user priorities have to be considered while opting for any solution, e.g. the Genetic Algorithm based solution may not be useful for some emergency situations where a quick convergence is required. At the same time, when there are no real time requirements Genetic Algorithms may be the right choice to go for. The best possible solution can be achieved by a technique that has inherent properties of learning, while characterizing the system performance with data obtained from observations performed by the system – CR. A brief description of the solutions used is provided in this Sub-section.

Reinforcement-Based Learning: It is represented as a directed graph, where the degree of success is attained in achieving the system short term and long term goals. Based on degree of success, the action causing success is assigned a reward weight to keep a track in case if similar inputs are encountered again.

Temporal Difference: It builds the state representation *on the fly* and does not require the back-propagation used in the reinforcement learning.

Simulated Annealing: The search space is explored by one solution at a time. It is inspired by physics. Another such single solution at a time based approach is Hill Climbing.

Collective Intelligence: We have already given the example of Swarm Intelligence and Ant Colonies of this biologically inspired approach.

Graph Algorithms: A network of graph is used to find the best possible graph that could solve the problem.

Markov-Decision Algorithms: The previous steps are considered to come up with the best solution in the next step. The previous steps' actions and the results to those actions are taken into account. Thus, we can say that this approach has inherent learning mechanism to some extent.

Game Theory Simulations: The basic modeling notion is a *game*, which includes: a set of players, actions for each of the players, principles for determining outcome of actions chosen by players, preference for the players and finally the rules. The players adapt their strategy and choose the actions in an attempt to maximize their outputs. This approach is gaining more and more popularity in the CR domain.

Till now, we have reviewed several methods that can be used to incorporate *learning* in the CRs. We have seen from the basic methods to highly advanced meta-heuristic based methods that there is no single approach to learning that will address all aspects of CR, e.g. computational efficient resources have timing constraints and vice-versa.

We decided in the favor of Artificial Neural Networks (ANN). We define the structural architecture and the working principle in this Sub-section. The specific neural networks that we have used, their architecture and the reason to pursue them as solution are addressed in Chapters 5 and 6.

4.6 Artificial Neural Networks:

Artificial Neural Networks, to some extent like GAs, also rely on the reinforcement of a decision or selection based on the actual result or outcome of a decision. A typical Neural Network has an input vector and an output vector. The middle layer, called as hidden layer, links the input and output values by processing them according to some rule. It propagates the input vector from input layer to output layer through a set of connections across different neurons of different layers. Suppose that each node in the neural network accepts an input vector, A , and applies a weight vector, W , to perform the propagation of the input value along the link to the adjacent layer in the network. This propagation is typically tempered by a constant or bias value, b . This bias value is controlled by the user, so that a known input vector may produce the desired output value. Thus, for an individual node, the propagation output, A , of a given node, j , can be expressed as shown in equation (3.1).

$$A_j = \left[\sum_{i=0}^n a_i w_i + b_j \right] \quad (3.1)$$

When the output value A_j is greater than some threshold value, T_i , the value is propagated along the output, as illustrated in Figure 11. Learning, within the context, of a cognitive radio, involves the adjustment of the threshold value, the bias value, or the weight associated with a node. This adjustment depends on the purpose for which the neural network is used for.

A neural network is actually a collection of individual nodes which are organized together in a multilayer fashion, as illustrated in Figure 12. The neural network has a set of input points. The values sensed or input through these points is propagated forward, through the middle layer (also referred to as hidden layer), to a set of output points. The output points activated by the forward-propagation are then compared with the actual value (i.e. anticipated versus actual). If there is a match, the path followed to arrive at the output point is followed through back-propagation, and the intervening nodes and paths are reinforced for that particular output joint.

Learning within a neural network requires feedback that allows the network to compare the expected output value associated with a set of input data against the conclusion reached by the neural network.

This forms the back-propagation. As the vectors of input values are applied to the system, the data propagates through the intermediate layers to the output. The expected output is mapped onto the output vectors. Those elements in an output vector whose value matches the applied expected value are reinforced by back-propagation. Thus, the intermediate layers that contributed to the propagation of data resulting in the correct (i.e. expected) output values are reinforced. Reinforcement may be performed through increasing weights, of the nodes involved in the propagation from the input data to the correct output or conclusion.

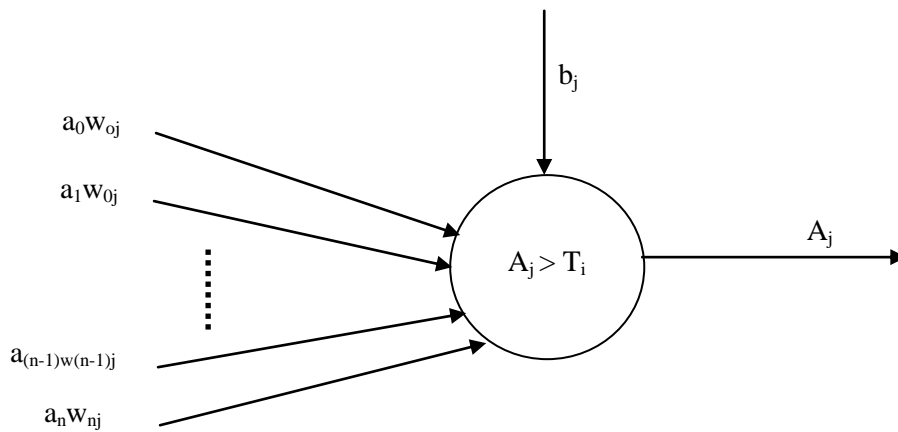


Fig. 11. Neural Network node illustration. Shown are the input vector A (with elements $a_0 \dots a_n$); weight matrix W (with elements $w_{0j} \dots w_{nj}$); bias value, b ; and output.

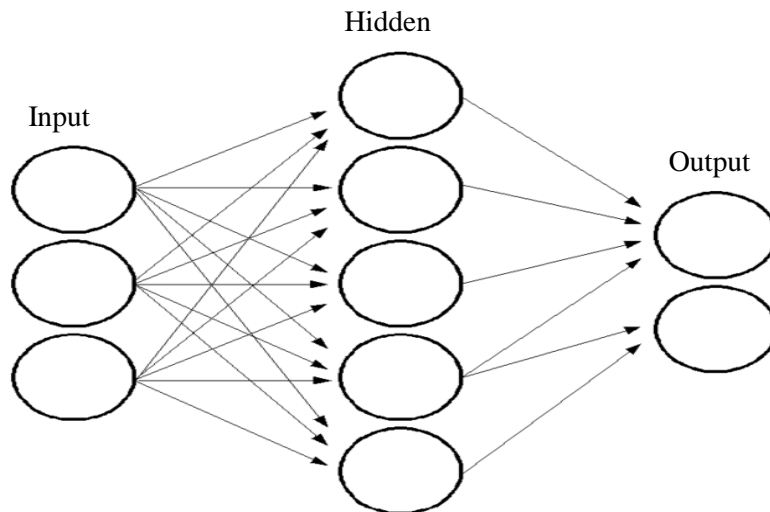


Fig. 12. Artificial Neural Network

This reinforcement of neural links increases the probability that the same input values will result in the same propagation to the correct output values. Those output values that did not match the expected output value are weakened, thereby decreasing the probability that they would be applied again given the same set of conditions.

We propose the learning schemes based on Artificial Neural Networks for the two purposes:

1. *Recognizing* a Standard in multi-standard CR Transceiver.
2. Spectrum Evolution *Prediction* enabling the SU to exploit the future available opportunity.

5. Major Cognitive Radio Projects and Achievements:

The main objective of this section is to have a review of important projects recently completed that concentrate on Cognitive Radio Engineering for new wireless communication paradigms. We have selected the following three projects, however there are numerous other projects as well that have contributed to this domain.

5.1 IDROMeL [37]:

IDROMeL is French acronym for an open platform for prototyping of Advanced Software Defined Radio and Cognitive Radio techniques. The target of this three year project (2006-2009) was to design an FPGA- based prototype of a multimodal MIMO open platform, to explore the CR scenario. The seamless mobility in a heterogeneous network was the aim so as to incorporate the ubiquitous connectivity. This was achieved by making two different Radio Access Technologies (RATs) to communicate with each other. The example RATs were UMTS and WiMAX to stress on the fact of different QoS, frequency bands and bandwidths. The communication in different bands and different waveforms are supported. The RF capabilities objectives (200 MHz – 7.5 GHz) allow the support for Standards: 802.11, UMTS, MC-CDMA, GSM, DVB-T, GPS, etc. The maximum bandwidth supported is 20 MHz. At the same time, each RAT was reconfigurable, in terms of its physical layer parameters. The handover between the two waveforms observes no degradation. This is achieved by sending the data flows from and to the terminal through two radio interfaces, simultaneously. The second waveform is installed on the terminal, while the first one is already running. After the communication flow starts operating on the second waveform, the first one is removed from the terminal. The platform components are: flexible baseband processing, MAC design running on Linux RTAI host PC to support the handover described above, a MAGALI (originally FAUST chip) Network on chip implemented in a 65nm technology from ST Microelectronics and partially reconfigured FPGA. The MAGALI chip has heterogeneous hardware blocks with both: generic and specialized functions. The features of the platform are extended by the partial reconfiguration of FPGA, which brings the highest flexibility to the hardware domain. The block diagram showing the components is shown in Figure 13. The reconfiguration of FPGA leads to few microseconds to change an IP, meeting the real time constraints of this technology. The transmitted power is comparable to existing GSM terminals (+21 dBm). The noise at the receiver side has the range of 8 to 12 dB. The baseband unit has seven processing units which are interconnected through an Advanced VCI crossbar interconnect. On

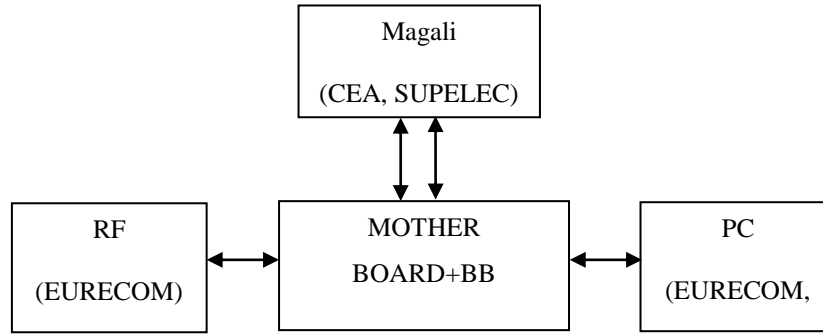


Fig. 13. Hardware Architecture of IDROMeL [37]

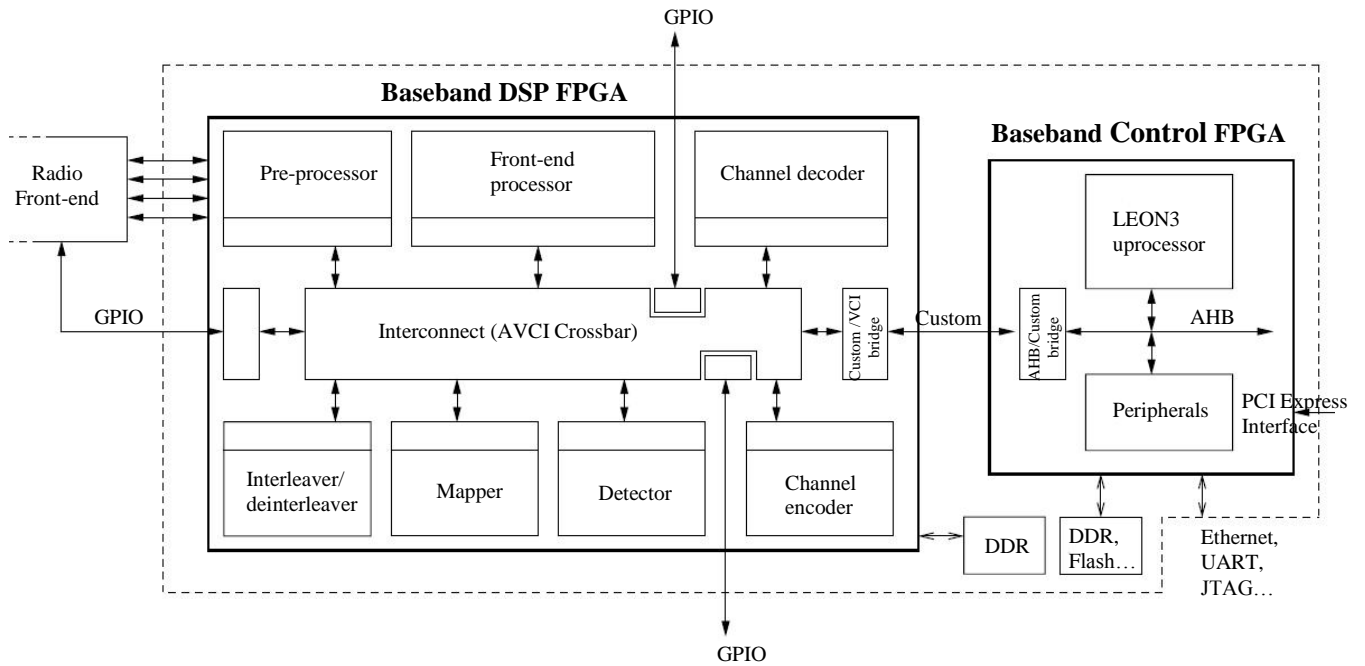


Fig. 14. Baseband Processor Architecture of IDROMeL (taken from [37])

transmission side, the D/A converters provide 14-bits resolution with sampling rates up to 128 Ms/s. On receiver side, the A/D converters provide 12 bits of resolution with sampling rates up to 64 Ms/s and usable analog bandwidths up to 100 MHz for IF sampling. The Base-band architecture and components interconnection is shown in Figure 14. The different processing units and their functionalities are:

A pre-processor that embeds a quadrature offset compensation unit, a DCO and programmable input-output FIFOs for synchronization.

A front-end processor implementing FFT.

A generic mapper and detector for different modulation schemes.

A generic channel coder that implements convolutional or cyclic codes.

A generic channel decoder for Viterbi Decoding.

A generic interleaver with rate matching, repeating and puncturing capabilities.

5.2 GRACE [38]:

The GRACE (Gestion du Spectre et Radio Cognitive) was a French national project spanned over three years (2006 -2009) concentrating on the definition, the evaluation and the demonstration of techniques of exploration of the spectrum in order to improve spectrum management. The project consisted of following four phases:

- **Scenarios, Decision and Context:**

This task defined the scenario comprising of radio and physical layer aspects apart from network and user management aspects.

- **Sensing and agility:**

This portion was about the study of algorithms and systems that consider the following three functionalities:

1. Sensing of RF environment.
2. Waveform generation in RF context.
3. Evaluation of interference caused by cognitive terminal to other active users.

- **Opportunistic Terminal Architecture:**

This part was meant for the consideration of the implementation constraints in terms of evaluating the integration complexity with respect to various spectrum management scenarios of an opportunistic terminal. With this knowledge, new architectural solutions for cognitive terminals were intended to propose followed by transmitter architectures for filter banks that led to concrete results. Finally, this phase identified the technical limitations that are needed to overcome.

- **Demonstration:**

This phase provided a feasibility of proposed solutions and suggested confrontations with the reasons. Unprocessed signals were provided to other parts of the project, for analysis, by this phase. The demonstration was carried out on current standards, e.g. IEEE 802.22 in order to exploit such standards to make the frequency usage more flexible.

5.3 End to End Reconfigurability (E2R) [39]:

The basic aim of this project was seamless reconfiguration. This project consisted of three phases with the first phase started in 2004. It had 32 partners that included CEA and Supélec. In this project, the spectrum and radio resource efficiency is increased by ambient intelligence vision. This ambient intelligence was realized by cross-disciplinary algorithms, inspired by physics (Simulated Annealing),

biology (Genetic Algorithms), demographic studies (Swarm Intelligence), Gaming Theory and Markov Decision process. The important achievements were the development of flexible guard bands, Joint Radio Resource Management (JRRM) Manager, Spectrum Efficiency Concepts, Cognitive Pilot Channel (CPC), development of Configuration Management Module (CCM) and the Execution Environment (EE). The objective of all these achievements was to devise an architectural design of a communication system that can offer a heterogeneous choice to user with respect to application and service providers, operators, manufacturers and regulators. The following key challenges were addressed with the help of eight technical work packages, in this project:

Design generic system mechanism that implements seamless experience management. These mechanisms build on, and are compatible with, legacy system management solutions in a multi-access situation.

Evolve the equipment management methods to include them as part of E2R architecture building on, and compatible with, legacy management solutions.

Develop cognition-based mechanisms by efficient spectrum, radio/equipment resource utilization, thereby facilitating efficient access to resources where there are multiple owner situations. There was a very close association with the regulation authorities, which enabled development of technologies for flexible spectrum resources and their associated usage, equipment circulation and security issues. The outcome to the regulatory discussions was intended to lead to a simplified framework that could allow flexible assignment and facilitating the spectrum use through techniques implementing an optimized usage. The objective to facilitate simpler and more market aware flexible spectrum management was exploited to operate where users will access unlicensed spectrum.

6. Conclusions:

This chapter defined and explained the Cognitive Radio notion and reviewed the important research challenges in its realization. Furthermore, the efforts to address these identified important research challenges are reviewed. We also give a brief overview of different candidate machine learning techniques for cognition incorporation in radios. Important research projects and their achievements are summarized so as to paint a better picture of the current scenario of Cognitive Radio. A brief summary of architecture evolution of cognitive radio is also included. This chapter shows that the potential for cognitive radio to make a significant difference to wireless communications is immense. We showed with the help of different projects that CR is an emerging research area that requires interdisciplinary collaboration and an overhaul of wireless systems design, performance evaluation, network operation and regulation.

The next Chapter treats the SDR waveform components design on a general purpose platform. As we

have introduced The International Technology Roadmap for Semiconductors (ITRS) prediction that out to 2017, software design productivity will fall behind hardware design productivity, in Chapters 1 and 2, we explain in detail our methodology to address this very challenge in the Chapter 4.

Chapter 4:

Embedded System Implementation and Optimization of SDR.

1. Introduction:

A challenge of Software Defined Radio lies in the implementation on embedded platform. Performance requirements under resource constraints make SDR the most difficult task. As the notion of SDR refers to a radio that can be reprogrammed via software, it guarantees a longer service life time in applications where changing waveforms on already acquired SDR equipment is required. Followed by the first working SDR effort, SPEAKeasy [3], a remarkable milestone in recent times is the first ever open source software Vanu Software Radio [5] engineering. Today SDR is given a more pragmatic interpretation which means that large and considerable parts of waveform are defined in software. This leads to the flexibility to change the waveform within certain bounds as given by the actual system. A rapid evolvement of communication standards makes it important to replace the base stations at regular intervals. SDR based embedded solution replaces this costly replacement by software upgrades, which is more cost effective. Although SDR adoption is not that widespread as it was predicted back in 2002: *By 2006 the adoption of SDR in commercial mobile terminals would have widespread adoption and movement to SDR as baseline design [20]*, there have been many success stories that tell us that we are moving closer to the wide adaptation of SDR based solutions in commercial products such as smart hand held devices. The Vanu Software Radio [5] *Anywave Base Station* can be quoted as the most significant example, as it is the first ever FCC-approved Software Radio [40]. In the case of applications requiring high volume, the preference goes to hardware based traditional radio architecture. At the same time the interoperability critical conditions are not met by these traditional implementations. This calls for an embedded solution based on general purpose processors. At the same time, the case when the lifetime of a product exceeds than the device with which the communication link has to be established and most importantly for wireless development issues, we need to switch to a general purpose based software design radio solution. This can be justified by seeing the latest trends in an increase in processing performance and power efficiency.

In such cases, general purpose processors based software radios are more capable than their competitive counterparts. In many other applications, such as SDR based CR solution, data acquisition in fields including magnetic resonance force microscopy, aeronautical applications, etc.. the success key for SDR based solution is reconfigurability, which is achieved in the highest degree in the general purpose processors. Furthermore, the *on the fly* optimization is also addressed in the most efficient way, together with being cost effective.

With this background, in this Chapter we set out to demonstrate the capability of general purpose solution as in terms of enabling portable communication offering a superior flexibility and advanced functionality.

2. SDR Embedded Implementation Efforts:

Since Reed [41] and Cummings [42] described the transition of FPGAs from Application Specific Integrated Circuits (ASIC) prototyping to embedded products, there have been some amount of research work that addresses SDR implementation on embedded platform with FPGA. Fifield [43] created an standalone FPGA design implementing a general purpose Orthogonal Frequency Division Multiplexing (OFDM) transmitter for software defined radio that supports basic 802.11 a/g transmission and allows OFDM parameters to be changed dynamically. Revé [44] discussed implementation of a Middleware, especially designed for SDR applications, called Platform and Hardware Abstraction Layer (P-HAL) when applied to hardware devices. Minden [45] presented the Kansas University Agile Radio (KUAR) platform and claimed it to be a low cost, flexible RF, small form factor SDR implementation that is both portable and computationally powerful. Schelle[46], using Xilinx's ISE and EDK 8.2 tools, showed the results of partitioning and placement of a SDR transmitter onto a Network on chip (NOC) architecture using a FPGA, thus exploiting reconfigurable hardware. In [47] Dikmese has implemented adaptive antenna array software radio beam formers on Virtex II Pro FG456 Protoboard after converting the VHDL code into a bit file using Xilinx ISE 7.1. The goal of multimode systems is realized by The Center for SDR Aalborg University, Denmark [48] by implementing a bi-standard (UMTS and WLAN) SDR receiver using Xilinx FPGA Virtex- 4. Myler [49] configured the hardware platform and software platform for cognitive radio equipments. In [50] NASA defines an open hardware architecture, Space Telecommunications Radio Systems (STRS), that abstracts functionality away from specific hardware devices through the hardware and software interfaces enabling greater use of design. Tachwali in [51] presents a BPSK Transceiver centered at 400 MHz that provides a framework for designing an adaptive digital transceiver on hybrid SDR platforms. Rakhshanfar [52] carried out a demonstration test bed implementation of Global System for Mobile Communications (GSM) receiver based on the idea of a flexible real-time software

radio using a personal computer (PC) and single FPGA. Recently, the latency, CPU load and memory utilization of OSSIE are estimated in [53] by G.Abgrall. Y.Lin in [54] presents a design study for a performance and power efficient programmable architecture; four processor core Signal-processing On-Demand Architecture (SODA) and demonstrates the flexibility by evaluating different high-end wireless protocols (W-CDMA and 802.11a). In [10], J. Glossner discusses the trends in inherently convergent platforms by presenting the Sandbridge state-of-the-art platform that is capable of :

1. Executing DSP, embedded control and Java code in a single compound instruction set optimized for handset radio applications.
2. Implementing multiple communication systems on a single SDR chip.

He further explains in [11] the Sandbridge SB3011 platform architecture that tells the implementation details of low power architecture that comprises of micro-architecture, logic and circuit design and software tools. He provides results for UMTS, DVB-H, WiMAX, WiFi and NTSC video decoding. B.Bougard in [55] presents the design of a hybrid Coarse-Grained Array Single Instruction Multiple Data (CGA-SIMD) baseband processor for SDR that achieves a clock frequency of 400 MHz in worst case conditions and occupies 5.79 mm². As an application case study, the processor is shown to be able to execute 20 MHz 2 x 2 SDM-OFDM baseband processing, achieving 100 Mbps + throughput, consuming 220 mW. In [56] Ng demonstrates a practical use of a complete Electronic System Level (ESL) design flow from high level virtual platform modeling to HW/SW co-verification of a large scale SDR SoC design. He co-emulates the design in two independent tool vendors, CoWare's SystemC and the mentor's emulator simulation environment. Addressing the relatively less researched topic of protocol processors executing upper layer protocols, H.Lee in [57] proposes a dual-processor platform with two GPPs: a conventional main processor, OPENRISC1200, and a simple low power supplemental processor, Microchip's 16F84. This platform successfully addresses the trade-off between flexibility and scheduling and thus maintains programmability at low power, apart from meeting real time deadlines of a SDR terminal. Y.Lin in [58] has designed a SPIR, a hierarchical dataflow programming model, to model SDR application. He has presented a coarse-grained dataflow compilation strategy that assigns SDR protocol's DSP kernels onto multiprocessors to determine an execution schedule that meets a prescribed throughput. As a case study, W-CDMA wireless protocol is used. Schiphorst in [59] describes a SDR testbed for wireless LAN standards. He implements the physical layer of HiperLAN/2 standard in real time software and baseband experiments have verified his system. However, the most demanding parts i.e. FEC coding/decoding have not been implemented.

This all review of related work shows that there have been significant efforts meant for SDR embedded implementation. However, technological advances in the coming years will push communications hardware into obsolescence at an even quicker pace. As a proof, The International Technology Roadmap for Semiconductors (ITRS) predicts that out to 2017, software design

productivity will fall behind hardware design productivity, due to inconsistency between software and hardware advancements [60]. Thus a challenge lies in the inconsistency between software and hardware advancement. For example, multi-core processor architectures and embedded multiprocessors-system-on-a-chip (MPSOC) are fueling significant improvements in performance while continually reducing their size and resource requirements. Unfortunately software is not keeping pace as indicated in Figure 15. This directly limits the capabilities of software applications, such as SDR, where functionality is kept in hardware, thereby prolonging the exposure to rapid hardware obsolescence. This chapter will discuss our approach to address this challenge.

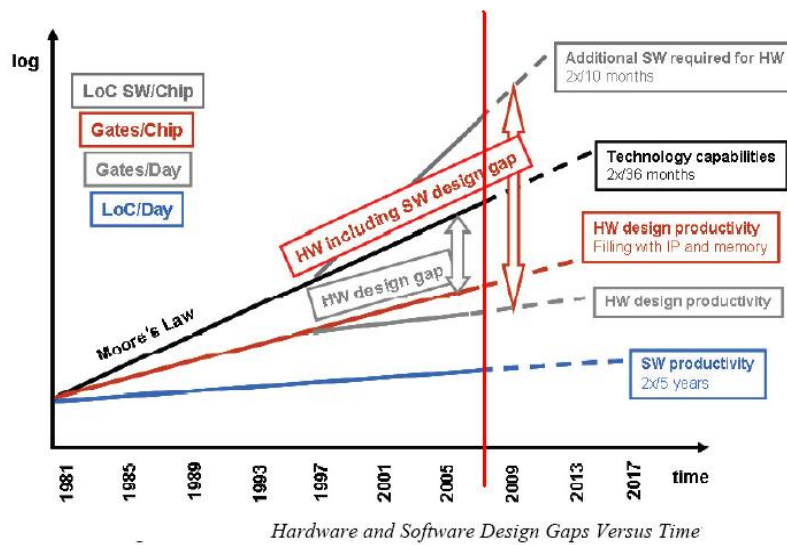


Fig. 15. ITRS Roadmap

3. MPSoC and FPGA Technology:

MPSoC have been designed to satisfy the requirements of embedded applications. They are actually the result of an aggregation of System on chip and traditional multiprocessors. The three most important features that these embedded applications possess are: high performance, real-time and low power. Higher performances are always required by consumers. This encourages the research and development of high-performance platforms that can meet new requirements and new standards. Higher performance means more computations and more complex algorithms that cannot be realized by simple hardware or single processor SoCs. MPSoCs have been designed with this background [61].

Real time computing is not only high performance computing. In the usual commonplace programming, we have to consider the speed and execution requirements, but the deadlines are not taken that seriously. However, in embedded systems environment, we have not only to cope up with

performance requirements, but have to meet the already decided deadlines. In this respect, MPSoC architectures have to be predictable to some extent, in the sense that applications should be facilitated to run with some predictable performance from the MPSoC architecture.

Another important constraint is the power consumption in the design of any MPSoC. Power constraints are more rigid in an MPSoC environment as compared to traditional supercomputer system or desktop computer system. Lower power consumption can extend the life of the battery in the case of battery operated MPSoC, apart from limiting the energy provided by battery required MPSoC design. Another possibly is the reduction in energy usage. In non-battery operated devices, low power requirement is there for chip heat and cost considerations. It is because of these constraints that MPSoC's power and energy constraints are required to be tackled at all the levels of abstraction.

The next section describes the target multi-core single chip platform meant for embedded implementation of SDR algorithms. The cognitive algorithms that we will describe in Chapter 5 are also implemented on the same target multi-core chip.

4. The Multiprocessor Platform:

The multicore implementations can be divided in two categories (1) general purpose, (2) application specific. ARM ARM11MPcore[62], the MIPS MIPS32 1004 Core [63] and the Renesas/Hitachi SH-X3[64] belong to the first category. Texas Instruments TMS320C6474/TMS320VC5441DSP [65-67], Freescale QorIQ P4080[68] and the ToshibaVenezia multicore [69] are in the second category. Our target 16 Processing Elements (PE) MPSoC with four DDR memory banks that we will describe in this Section falls in the first general purpose category. The motivation for general purpose MPSoC solution for SDR embedded waveform implementation comes from the fact that these general purpose MPSoCs have become the standard for implementing high adaptability agile frequency operation devices such as SDR and CR. Another motivation to choose a general purpose reconfigurable solution lies in the fact that apart from our SDR waveform resources: FFT, Viterbi Decoding, Filter, Algebraic, Modulation and Demodulation Functions, our Artificial Neural Networks based cognition algorithms: LVQ, SOM and MLP, may benefit equally from the reconfigurability offered by such general purpose platforms. We will explain these cognitive algorithms in detail in Chapters 5 and 6.

The SDR waveform components that we will define later in this chapter are mapped on our target general purpose multiprocessor embedded system realized on Xilinx FPGA Virtex-4 FX-140 FPGA chip shown in Figure 16, using Xilinx EDA tools [61] using our designed parallelization strategy. All the Processing Elements and memory controllers are connected with a scalable On-Chip Network (OCN) developed with Arteris Danube library [70]. We describe the design and technology



Fig. 16. FPGA board

constraints and how it affects the architecture of the target MPSoC designed for Cognitive SDR waveform.

4.1 External Constraints:

The external pin connection constraints of FPGA chip shown in Figure 16 reduce the possible number of external DDR modules to 4, as shown in Figure 17.

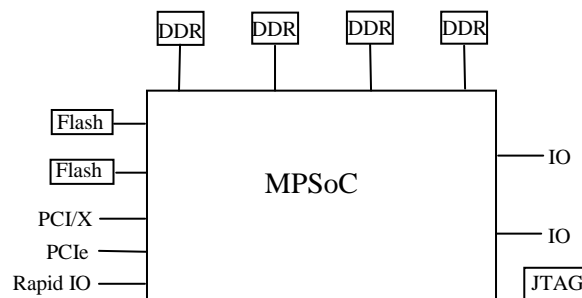


Fig. 17. MPSoC Target External Connections [61]

With the shown constraint of 4 external DDR, the designed MPSoC architecture [61] is organized around classical dance hall architecture in contrast to other available architectures. The memory banks are on one side and processors on the other side of the Data NoC, as shown in Figure 18. A NoC is essential in this case as 16 implemented processors are connected on the communication medium. As shown in Figure 18, Data NoC is connected to four DDR controllers, which in turn connect to four off-chip 256 MBytes DDR memory (totally 1 GBytes). PE tiles are connected to Data-NoC and Synchronization NoC through OCP-IP interfaces.

One 64 KBytes shared on-chip memory is attached to the synchronization NoC, which establishes a

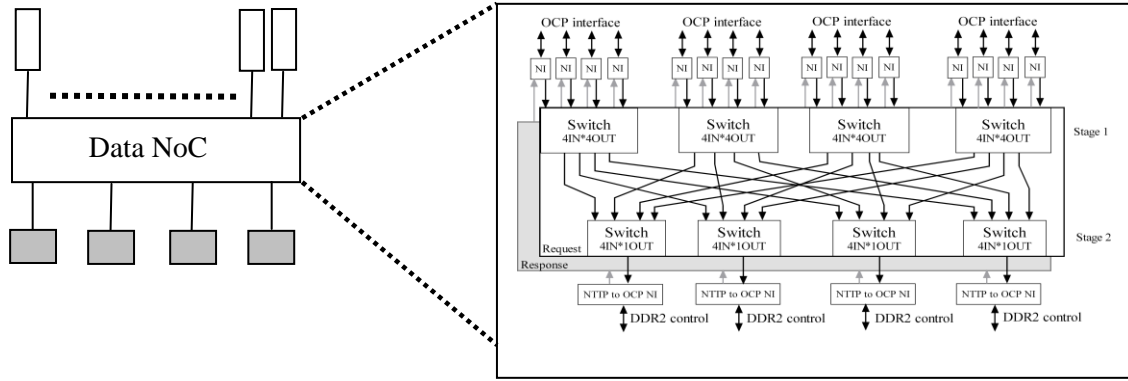


Fig. 18. Block Diagram of designed MPSoC Architecture [61]

synchronization media for the 16 PE tiles.

The execution frequencies of Data NoC and Synchronization NoC are different due to different pipeline strategies, arbitration settings and configuration of input, output numbers. Data NoC runs at 200 MHz whereas Synchronization NoC runs at 250 MHz. The frequency of the processor used is 130 MHz.

4.2 MPSoC Architecture:

The Virtex-4 architecture adds an additional feature described in Figure 19.

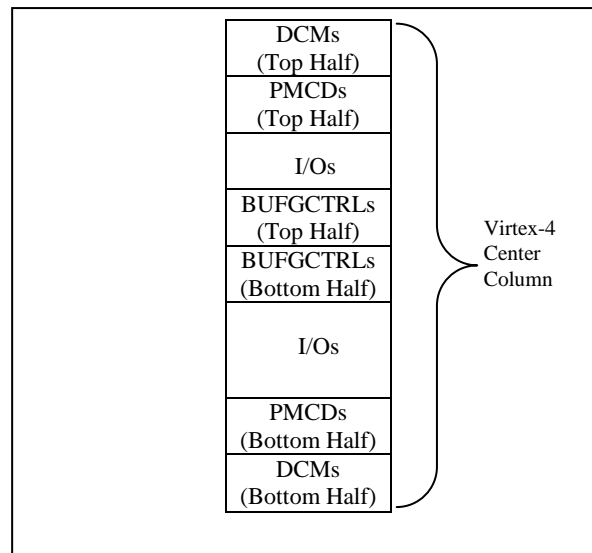


Fig. 19. Virtex-4 Internal Architecture

The Digital Clock Managers (DCM) for clock tree and I/Os are placed in the center of the chip by the central column. Any IP not accessing should then be placed at the periphery of the layout. This is the case of processor IPs when they are accessing I/Os such as DDR controllers, so should be placed in the center. Therefore, we can justify the choice of MPSoC architecture shown in Figure 18.

4.3 Processing Element:

The processors used on FX-140 chip are Xilinx Microblaze [71]. Each Processing Element (PE) can independently run its own program code and operating system. These Microblaze processor based PEs are connected to switches through Open Core Protocol to NoC Transaction and Transport Protocol (OCP-to-NTTP) Network Interface Units (NI). The OCP-to-NTTP NI or Master NI translates OCP to the OCN protocol: Arteris NoC Transaction and Transport protocol (NTTP). This switching system has connection to four NTTP-to-OCP NIs (Slave NI), which in turn has connection to the respective DDR2 memory controller. Each DDR2 controller controls an off-chip DDR2 memory bank (256 Mbytes). The block diagrams of the implemented PE are shown in Figures 20 and 21.

In order to enhance the compatibility and to make the architecture re-utilizable, the OCP-IP standard is used for the connection of PEs and OCN. Having the benefit from the OCP standard, any processor possessing OCP interface can be easily connected to the system. The Xilinx processing soft-core Microblaze v.7.00 based computing system that is integrated as a PE in the FPGA design, is a 32 bit reduced instruction set computer (RISC) optimized for implementation in Xilinx FPGA, and the IPs of Microblaze processor and the memory connection are provided in the library of the FPGA design environment: Xilinx Embedded Development Kit (EDK).

Microblaze processor is implemented with Harvard memory architecture: instruction and data accesses

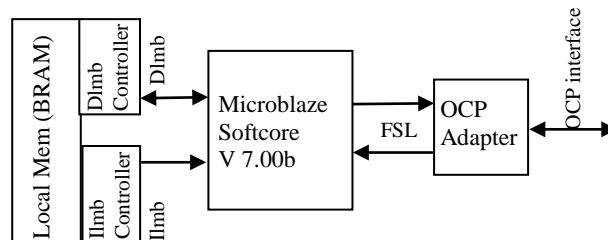


Fig. 20. Microblaze based Processing Element [61]

are done in separate address spaces and it is highly reconfigurable. It is a 32 bit either 3 stage (area optimization enabled) or 5 stage (area optimization disabled) architecture. It is provided as a part of Xilinx embedded design tool kit to fit into Xilinx FPGA and has an orthogonal and flexible reduced instruction set architecture (RISC). It has thirty-two 32 bit general purpose registers (numbered R0 through R31) and up to eighteen 32-bit special purpose registers (depending on configured options). It uses memory mapped I/O and instruction and the data cache can be configured to use 4 (software for a random access pattern) or 8 (software for a sequential access pattern) word cache lines. A set of other parameters can also be configured at design time to fit design requirements, such as number of

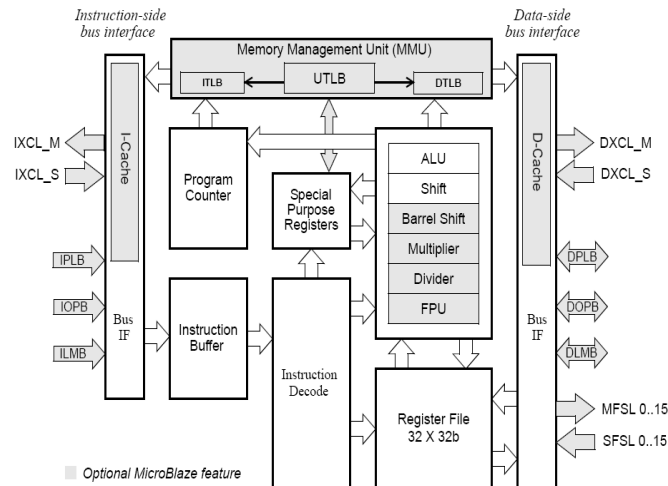


Fig. 21. Microblaze Core Block Diagram

pipeline stages, cache size, interfaces and execution units like: selectable Barrel Shifter (BS), Floating Point Unit (FPU), hardware divider (HWD), hardware multiplier (HWM), memory management unit (MMU). The configurability allows the user to trade-off features for size, in order to achieve the necessary performance for the target application at the lowest possible cost point. The performance and maximum execution frequency vary depending on processor configuration. For its communication purposes, Microblaze v7.00 offers a Processor Local Bus (PLB) interface and up to 16 Fast Simplex Link (FSL) interfaces which is a point to point FIFO-based communication channel. Highlights of PLB protocol include synchronous architecture, independent read/write data paths and split transaction address/data buses. The frequency of each PE tile is 130 MHz. Two memory controllers control 32 Kbyte BRAM based local on-chip memory. As OCP interface is not provided by Microblaze, an OCP adapter, which can translate FSL to OCP interface, is integrated in PE subsystem for the connection with OCN as shown in Figure 20. OPB and PLB devices can communicate by way of an OPB-to-PLB Bridge or a PLB-to-OPB Bridge. Device Control Register (DCR) bus is used for accessing control and status registers in various devices. It allows for register access to various devices without overloading the OPB and PLB interfaces. Because DCR devices are generally accessed infrequently and do not have high performance requirements, they are used throughout the reference design for functions, such as error status registers, interrupt controllers and device initialization logic.

4.4 OCN:

The On-chip Network connection system has been developed with the Packet Transport Units (PTU) from the Arteris Danube Library. The packet transport portion is built by these PTUs for the NoC, comprising of a request network and a response network. The three-layered approach consisting of comprising transaction, transport and physical layers called as NoC Transaction and Transport protocol (NTTP) is adopted by all the PTUs. The conversion of OCP 2.2 protocol to NTTP protocol is done by OCP-to-NTTP NIs. Our OCN supports locked synchronization. *ReadExcursive* (ReadEX) and

Write or *WriteNonpost* commands are used by OCP initiator. This is done to do an atomic transaction: read-modify-write. It is used by NI that sends a Lock request packet after receiving the ReadEX command, which locks the path from master to the slave. During this whole locked period, the other Masters are prohibited to access the locked slave until the Master requesting ReadEX sends a Write or WriteNonPost command which ultimately unlocks the path.

The chip area consumed for different resources on our Xilinx Virtex-4 FX140 FPGA based chip is shown in Figure 22.

5. OSSIE Core Framework:

We have defined in detail, OSSIE open source software in Chapter 2. The software package includes a SDR Core Framework based on the JTRS SCA, tools along with signal processing components and waveforms (applications), device interface software and node configuration file for use with OSSIE. It runs on Intel and AMD based PCs and a later release including enhanced support for embedded as well as PC-based applications is also added. An explanation of OSSIE core framework and signal processing library follows:

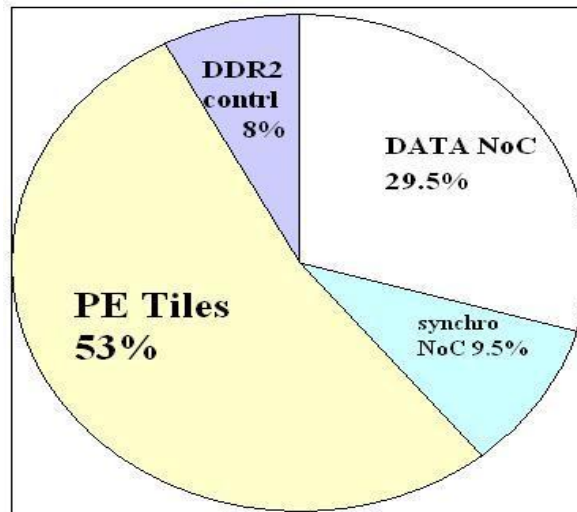


Fig. 22. Resource use in percentage.[61]

The OSSIE Core Framework defines the essential ‘core’ set of open software interfaces and profiles that provide for the deployment, management, interconnection and intercommunication of software application components in an embedded, distributed computing communication system. The OSSIE

Core Framework consists of the three interfaces as defined in Chapter 2, and a detailed list of classes in each interface can be found at [7].

5.1 OSSIE Signal Processing Library:

The Signal Processing Library (SigProc) is intended to provide a library of DSP algorithms commonly used within radio-communications. There is only one name-space *SigProc*. A brief description of important methods is necessary in order to understand the library. The file *Filter.cpp* implements the Polyphase Filter bank [72] and the filters supported are Square-root raised-cosine and Derivative. The pointer to character type *_type* determines the type of filter. A brief description of the functions is given in Table 2. The input parameters for *DesignRRCFilter* are *k* (samples per symbol), *m* (sample delay) and β (excess bandwidth or roll-off factor such that $0 < \beta < 1$). The function returns *h* (a pointer to filter coefficients) as well as an integer value describing the length of the filter. The length of a filter is given by equation (4.1) as,

$$h_{len} = 2km + 1 \quad (4.1)$$

The filter coefficients themselves are derived from the equation (4.2) as shown below,

$$h[z] = 4\beta \frac{\cos[(1+\beta)\pi z] + \sin[(1+\beta)\pi z] / 4\beta z}{\pi \sqrt{T[1-16\beta^2 z^2]}} \quad (4.2)$$

The function compensates for the two cases where $h[n]$ might be undefined in the above equation.

TransposeCoefficientMatrix() transposes *H* (filter bank coefficient matrix).

CalculateDerivativeFilterCoefficients() calculates derivative filter coefficients. The derivative of the template filter is approximated by (4.3).

Depending upon *_type*, the constructor of *FIRPolyphaseFilterBank()* calls the appropriate methods to calculate the filter coefficients for different filter types. *Npfb* determines the number of filters in filter bank. Filter coefficients are stored in two dimensional matrix, referred to as buffer. The matrix is of dimension *Nbfp* * *hlen*. *Modem.cpp* comprises of modulation and demodulation definitions shown in Table 2. The levels for each modulation/demodulation scheme are defined in *SigProc.h*. The method *rotate()* referenced by *DemodulateQPSK()* and *Demodulate8PSK()* that rotates a complex signal

$$\begin{aligned}
h(nT) &= \frac{\partial}{\partial T} h(nT) \\
hm(nT) &= h_{m+1}(nT) - h_{m-1}(nT) \\
h_0(nT) &= h_1(nT) - h_{m-1}(nT) \\
h_{m-1}(nT) &= h_{m-2}(nT) - h_0(nT)
\end{aligned} \tag{4.3}$$

counter-clockwise by θ is defined in class file *utility.cpp*. Also the *dot_product* definition is included in *utility.cpp* that calculates dot products between the length of a filter and the output sample; and is referenced by *ComputeOutput()*. *Randf()* is a uniform random generator whereas *Randn()* is a gaussian number generator. The other important classes include *scaling.cpp* and *sources.cpp*.

We started our work by the performance evaluation of SigProc on embedded platform that we describe in Section 6. The functions defined above can be classified in four classes:

1. Filter Functions.
2. Algebraic Functions.
3. Modulation Functions.
4. Demodulate Functions.

We show the important functions of these four categories of SigProc namespace in Table 2.

6. Performance Evaluation of OSSIE SigProc on Embedded Platform

ML-403 Board:

The ML403platform [73] provides an environment for developing embedded designs based on Virtex-4 FX FPGA. It has a 100 MHz Oscillator with 2 clock sockets, 64 MB DDR SDRAM, 64 Mb Flash is also included apart from 8Mb SRAM and 4kb EEPROM. It has a display of 16*2 characters LCD and connectors and interfaces include 4 SMA connectors, 2 PS/2 connectors, 2 audio(In/Out, Microphone/Headphone), 3 USB ports and other general purpose I/Os (LEDs and Buttons). Figure 23 is the block diagram of ML403 and Figures 24 and 25 show the components on the board. Microblaze processor, the most important part of the system is already explained in detail in Section 4.3.

Functions	Description	Description
Filter Functions	DesignRRCFilter	Calculates the length of filter and returns a pointer to the filter coefficients
	CalculateRRCFilterCoefficient	Calculates root-raised cosine filter coefficients
	CalculateDerivativeFilterCoefficients	Calculates derivative filter coefficients
Algebraic Functions	TransposeCoefficientMatrix	Transposes the filter bank coefficient matrix
	Dot_product	Calculates dot product between two floating point integer arrays
Modulation Functions	Modulation.	<i>ModulateBPSK, ModulateQPSK, ModulateQAM4, Modulate8PSK, Modulate16QAM and Modulate4PAM</i> respectively modulates a symbol into an I/Q pair for Binary Phase Shift Keying, QUadrature Phase Shift Keying, Quadrature Amplitude Shift Keying, 8-ary phase shift keying, 16 point Quadrature Amplitude Modulation and 4-ary Pulse Amplitude Modulation.
Demodulate Functions	Do_work	Detects the phase of the signal to be demodulated.
	Demodulation	<i>DemodulateBPSK, DemodulateQPSK, DemodulateQAM4, Demodulate8PSK, Demodulate16QAM and Demodulate4PAM</i> respectively demodulates the I/Q pair to get the BPSK, QPSK, QAM4, 8PSK, 16QAM and 4PAM symbol.

Tab. 2. Important Functions of SigProc namespace

ML403 Board is used to evaluate the execution time of SigProc defined in Section 5.1. The number of clock cycles for the radio-communication functions shown in Table 2 are calculated and shown in Table 3. Worst case execution time is taken into consideration while selecting input from set of all possible inputs.

Table 3 shows that as a result of this mapping, the filter functions are identified to be strong candidates for optimization so as to make them more performance efficient. One of the challenges in optimization is how best to customize the cache subsystem for improved performance with the algorithms. Our next step is to organize the cache size to maximize the performance of the OSSIE SigProc functions.

Studies have found that embedded processors power usage has been reduced by as much as 50% through cache optimization [74]. So, there has been significant work done for the hardware optimization including cache for the SDR software to be mapped on embedded platform. S. Kannan

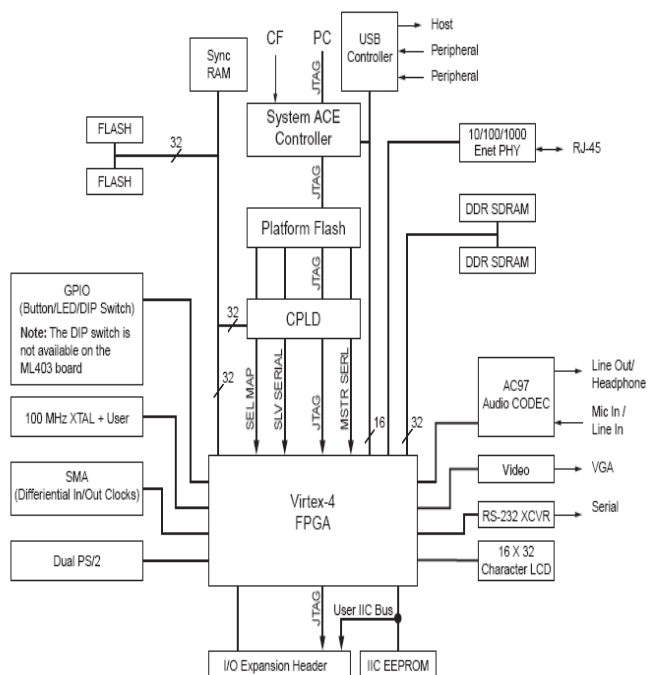


Fig. 23. Block Diagram of ML403

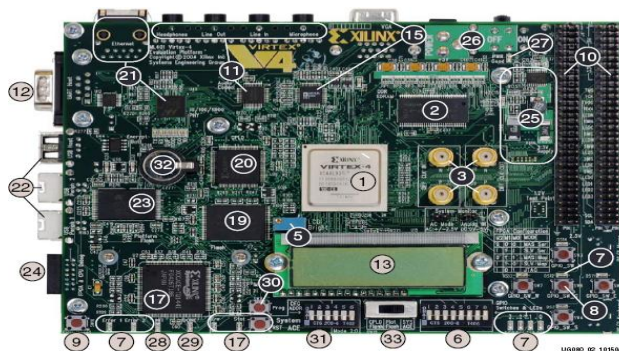


Fig. 24. Detailed Description of Virtex-4 ML403 Evaluation Platform Components (Front View)

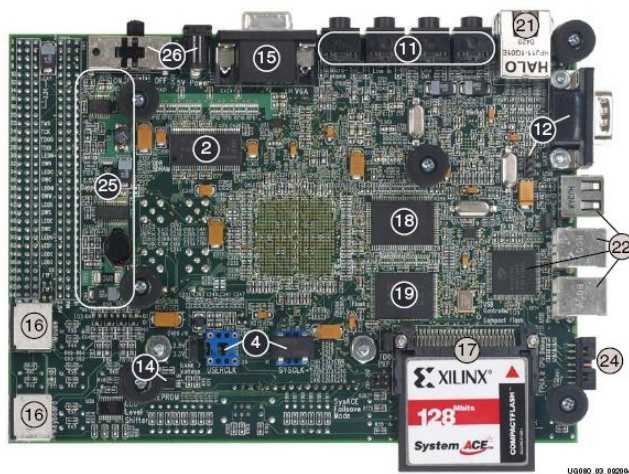


Fig. 25. Detailed Description of Virtex-4 ML403 Evaluation Platform Components (Back View)

Functions	Input	Clock cycles (worst case)
Filter Functions	DesignRRCFilter($k=10, m=1.5, \beta=0.3$) CalculateDerivativeFilterCoefficients($h_len=30, NPfb=38, N=30*38$) CalculateRRCFilterCoefficients($k=10, m=1.5, \beta=0.3, Npfb=38$)	DesignRRCFilter=928204 CalculateDerivateFilterCoefficients=35116101 CalculateRRCFilterCoefficients=40320
Algebraic Functions	TransposeCoefficientMatrix(No parametrs) Dot_product($N=100$)	TransposeCoefficientMatrix=564 Dot_product=146609
Modulation Functions	ModulateBPSK(symbol_in=1), ModulateQPSK(symbol_in=3) ModulateQAM4(symbol_in=3), Modulate8PSK(symbol_in=7), Modulate16QAM(symbol_in=15) Modulate4PAM(symbol_in=3)	ModulateBPSK=56, ModulateQPSK=72, ModulateQAM4=96, Modulate8PSK=78, Modulate16QAM=80, Modulate4PAM=69
Demodulation Functions	Do_work($I_in=9487, Q_in=3162$), DemodulateBPSK($I_in=-10000, Q_in=0$) DemodulateQPSK($I_in=-10000, Q_in=0$) DemodulateQAM4($I_in=7071, Q_in=7071$) Demodulate8PSK($I_in=7071, Q_in=7071$) Demodulate16QAM5($I_in=-3162, Q_in=3162$) Demodulate4PAM($I_in=-13416, Q_in=0$)	Do_work=2455, DemodulateBPSK=45 DemodulateQPSK=23507 DemodulateQAM4=70 Demodulate8PSK=24483 Demodulate16QAM=135 Demodulate4PAM=69

Tab. 3. Worst case execution time of SigProc Functions

proves in [75] that Write Through Data Cache Strategy is 30% more efficient than Write Back in the case of GSM/GPRS/EDGE algorithms implemented on hardware. He uses the AD6532 device, from the MSP500 digital baseband platform family. It must be noted that that the Microblaze implements Write through Data Cache Policy as well. In [76] Kim developed a reconfigurable module that performs as a function unit and a cache simultaneously, at the cost of 60 % area overhead. In the next sub-section, we propose the optimized cache size leading to minimum degradation of OSSIE radio-communication functions.

6.1 Microblaze cache:

Microblaze can be used with an optional cache for improved performance. The data cache has the following features:

- Direct mapped (1-way associative)
- Write- Through
- User selectable cacheable memory addresses range.
- Configurable cache size and tag.
- Option to use 4 or 8 word cache-lines.
- Implies Least Recently Use (LRU) replacement policy.

We have used the data cache by splitting the memory into two segments: a cacheable segment and a non-cacheable segment. The cacheable area is determined by two parameters: `C_DCACHE_BASEADDR` and `C_DCACHE_HIGHADDR`. All addresses within this range correspond to the cacheable address space. In this step of SigProc Filter Functions optimization, the cacheable address space is changed over and over again to get the least possible clock cycles. Rest all addresses are non-cacheable. The Microblaze data cache is configurable from 64 bytes to 64 kB. The data cache organization for Microblaze is shown in Figure 26.

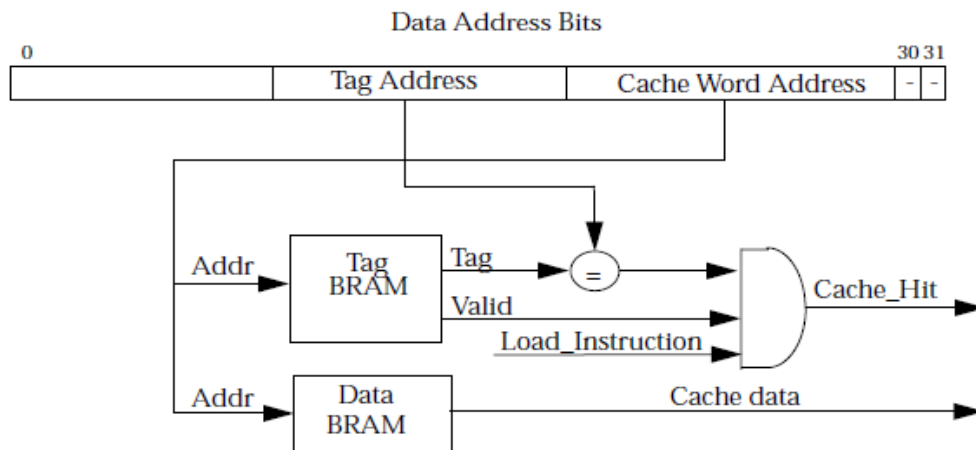


Fig. 26. Data Cache Organization for Microblaze

A load from an address situated in the cacheable range starts a check that determines if the required data is currently in cache. In case of a cache hit, the data is retrieved, otherwise if a cache miss occurs the required data is asked to be fetched at data CacheLink (DXCL), and the processor pipeline stalls till the associated cache line to the required address is returned from external memory controller. The Microblaze, in the created project, is configured to cache data over dedicated Xilinx CacheLink Interface.

The Cache is enabled using the appropriate functions by adding the header file *mb_interface.h*. The Cache Line length is chosen to be 4 throughout the experiments, whereas the BRAM is 64 kB throughout the project. The Data Cache is considered whereas the Instruction Cache is not considered. The most important design parameter is Cache Size as Cache memory has cost and space constraints, so the decision of how large a cache to implement in a system is critical. We have changed the Cache memory size for each filter function and the performance is evaluated in terms of clock cycles. The results for each cache size are shown in Table 4. The cache sizes are varied from 1 kB to 16 kB and for each cache size a new project is created to calculate the clock cycles so as to obtain performance evaluation. The Filter functions of SigProc are chosen as we had already identified them that they are more likely to be effected by the change in cache organization.

Cache Size	Clock Cycles for CalculateDerivativeFilterCoefficients	Clock Cycles for DesignRRCFilter	Clock Cycles for CalculateRRCFilterCoefficients
1 kB	9275	802864	35105096
2 kB	2864	802789	34100102
4 kB	2792	64514	34071389
8 kB	2544	58012	19535696
16 kB	2455	58012	9867848

Tab. 4. Filter Functions Performance Comparison for different Cache sizes.

The experimental results shown in Table 4 indicate that with the increase in cache size the performance is improved but after reaching a certain cache size, the law of diminishing returns comes into play and the performance improvement is not that significant. Keeping in consideration the inherent cost and space constraints of cache in an embedded environment and based on the results shown in Table 4, we propose the following cache size configuration for each filter function that will help in optimal utilization of SRAM.

Filter Function	Our proposed Cache Size
CalculateDerivativeFilterCoefficients	2 kB
DesignRRCFilter	4 kB
CalculateRRCFilterCoefficients	2 kB

Tab. 5. Our Proposed Cache Size for each Filter Function

7. OSSIE Signal Processing Function Performance Enhancements through Parallelization:

Addressing the ITRS Roadmap Prediction, our next step is to map the filter functions onto our multiprocessor embedded system that we defined in detail in Section 4. There has been some work on parallelizing applications on multi-processor SDR platform, e.g. the baseband processing in a space division multiplexing (SDM)-orthogonal frequency division multiplexing (OFDM) has been parallelized in [77] on a platform called ADRES [78]. P.J. Balister [79] tells about processor usage by porting the waveform in Figure 27 and getting the component processor usage to conclude the same result as ours.

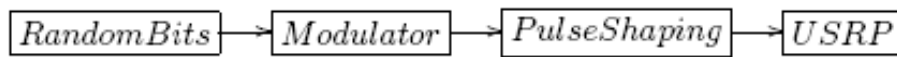


Fig. 27. Block Diagram for Random Data Transceiver

The results obtained for each component of the waveform shown in Figure 27 are shown in Table 6. P.J.Balister uses OMAP as target embedded system to port OSSIE.

Component	% Processor Usage
PulseShaping	50.3
USRP	3.8
RandomBits	2.0
Modulator	1.9
omniNames	0.1

Tab. 6. Component Processor usage

In Table 6, PulseShaping contains a FIR filter used for raised cosine pulse shaping. The chip we have used integrates sixteen Microblaze-based processing-element tiles that are connected through a Network-on-chip (NoC) configuration, as we have explained in detail in Section 4.

We visualize in detail our parallelization strategy for OSSIE Filter functions in Figure 28. The master processor executes various control tasks for multiprocessor environment. In our case, for each filter function (see Table 3) being parallelized, this master processor gives the launch signal to other processors via synchronized memory, by starting a timer. The slave processors, upon receiving

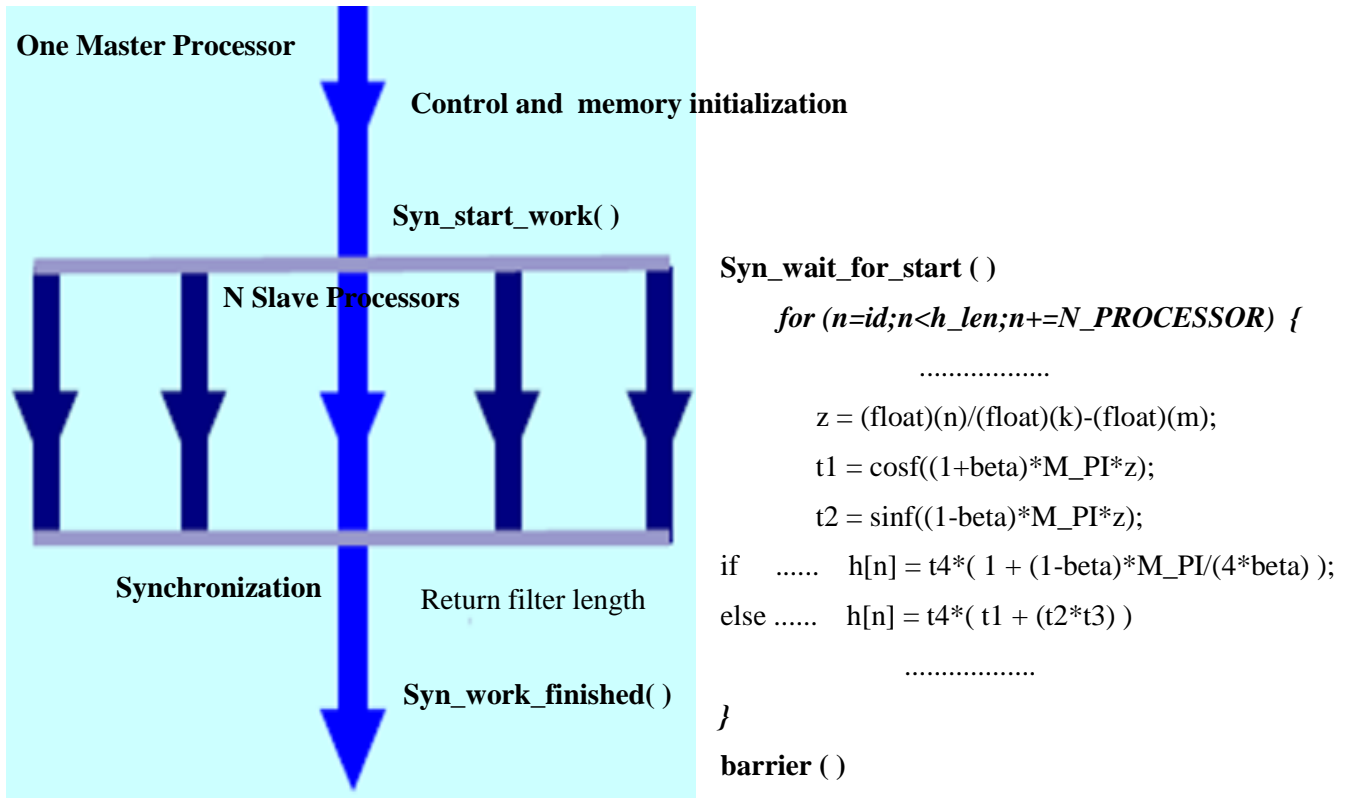


Fig. 28. Parallelization Strategy for OSSIE Filter Functions

the master processor start signal, begin calculating their respective filter coefficients. These processors accomplish this by splitting the loop calculating coefficients into N chains with N being the number of involved processors. Once all coefficients and the filter length for each respective function is calculated within each processor, the slave writes a flag into the synchronized memory to signal that it has finished its job. Once all slave processors have finished their tasks, the master processor captures the number of clock cycles. In the following subsections, we define the parallelization primitives and their usage to accomplish the designed strategy.

7.1 The Parallelization Primitives:

The parallelization objective is achieved with the help of following designed primitives:

- **Syn_start_work()**

Give slaves the command to start to calculate the respective coefficients.

- **Syn_work_finished()**

Check if slaves have finished calculating their share of coefficients.

- **Syn_wait_for_start ()**

Each slave waits for master to give this signal before starting and followed by this signal, it starts its assigned calculations.

- **barrier ()**

Once each slave is finished its assigned filter coefficient calculation, it uses this function to tell master that slaves have finished their assigned tasks. After all the slaves finish executing this function, then master captures the number of clock cycles.

7.2 Parallelization Description:

In this Sub-section, we describe the algorithm design of the three OSSIE Filter Functions, in the SigProc Library. We have already explained all the SigProc functions in detail in Section 5.1 of this Chapter. In this Section, we deal with the changes in the Algorithm Design made by us, so as to map them efficiently on our MPSoC. The important variables of these functions, together with our newly introduced variables for efficient parallelization of these functions are:

- **h_len:** Filter length = $2k+1$.
- **k:** samples per symbol.
- **m:** sample delay.
- **id:** identity of the specific processor that will carry out the task.
- **N_PROCESSOR:** Total Number of Processors (16 in our case).
- **h_len_p:** $(2k \cdot N_{pfb} \cdot m)+1$.
- **Npfb:** Number of Filters in Filterbank.
- **N:** Number of filterbank coefficients.

We change the loops of the functions in such a way that leads to efficient load-balancing across all the Processing Elements. As an example, in DesignRRCFilter function, the loop to calculate the filter length: *for (n = 0; n < h_len ; n++)* is changed to *for(n=id ; n < h_len ; n+= N_PROCESSOR)*. The variables used in the loops are already defined above.

We give the enhanced algorithm design and experimental results for achieved speed up in this Sub-section for:

1. DesignRRCFilter.
2. CalculateRRCFilterCoefficients.
3. CalculateDerivativeFilterCoefficients.


```

1. CalculateRRCFilterCoefficientsParallel(
2.     unsigned int id,
3.     unsigned int k,    // samples per symbol
4.     unsigned int Npfb,
5.     float m,          // delay
6.     float beta,       // rolloff factor ( 0 < beta <= 1 )
7.     float * h )      // pointer to filter coefficients
8. {
9.     unsigned int kp;
10.    unsigned int h_len_p;
11.    float h_scale = (float)(k);
12.    unsigned int n;
13.    float z, t1, t2, t3, t4    /*T(1.0f)*/;
14.    float T=1.0f;
        // Create over-sampled pulse
15.    unsigned h_len = 2*k*m*Npfb+1;
16.    kp=Npfb*k;
17.    h_len_p = 2*kp*m+1;
        // Calculate filter coefficients
18.    for (n=id; n<h_len_p; n+=N_PROCESSOR)
19.    {
20.        z = (float)(n)/(float)(kp)-(float)(m);
21.        t1 = cosf((1+beta)*M_PI*z);
22.        t2 = sinf((1-beta)*M_PI*z);
        // Check for special condition where z equals zero
23.        if ( n == kp*m ) {
24.            t4 = 4*beta/(M_PI*sqrtf(T)*(1-(16*beta*beta*z*z)));
25.            h[n] = t4*( 1 + (1-beta)*M_PI/(4*beta) );
26.        } else {
27.            t3 = 1/((4*beta*z));
28.            float g = 1-16*beta*beta*z*z;
29.            g *= g;
        // Check for special condition where 16*beta^2*z^2 equals 1
30.            if ( g < 1e-3 ) {
31.                float g1, g2, g3, g4;
32.                g1 = -(1+beta)*M_PI*sin((1+beta)*M_PI/(4*beta));
33.                g2 = cos((1-beta)*M_PI/(4*beta))*(1-beta)*M_PI;
34.                g3 = -sin((1-beta)*M_PI/(4*beta))*4*beta;
35.                g4 = -2*M_PI;
36.                h[n] = (g1+g2+g3)/g4;
37.            } else {
38.                t4 = 4*beta/(M_PI*sqrtf(T)*(1-(16*beta*beta*z*z)));
39.                h[n] = t4*( t1 + (t2*t3) );
40.            }
41.            h[n] /= h_scale;
42.        }
43.    }

```

Fig. 29. CalculateRRCFilterCoefficients Parallelized Algorithm Design


```

1.  int DesignRRCFilterParallel(
2.      unsigned int id,
3.      unsigned int k,    // samples per symbol
4.      float m,          // delay
5.      float beta,       // rolloff factor ( 0 < beta <= 1 )
6.      float * h )      // pointer to filter coefficients
7.      {
8.          unsigned int h_len;
9.          unsigned int n;
10.         float z, t1, t2, t3, t4 ;    /*T(1.0f)*/
11.         float T=1.0f;
12.         h_len = 2*k*m+1;
           // Calculate filter coefficients

13.         for (n=id; n<h_len; n+=N_PROCESSOR)
14.             {
15.                 z = (float)(n)/(float)(k)-(float)(m);
16.                 t1 = cosf((1+beta)*M_PI*z);
17.                 t2 = sinf((1-beta)*M_PI*z);
           // Check for special condition where z equals zero

18.                 if ( n == k*m ) {
19.                     t4 = 4*beta/(M_PI*sqrtf(T)*(1-(16*beta*beta*z*z)));
20.                     h[n] = t4*( 1 + (1-beta)*M_PI/(4*beta) );
21.                 } else {
22.                     t3 = 1/((4*beta*z));
23.                     float g = 1-16*beta*beta*z*z;
24.                     g *= g;
           // Check for special condition where 16*beta^2*z^2 equals 1

25.                     if ( g < 1e-3 ) {
26.                         float g1, g2, g3, g4;
27.                         g1 = -(1+beta)*M_PI*sin((1+beta)*M_PI/(4*beta));
28.                         g2 = cos((1-beta)*M_PI/(4*beta))*(1-beta)*M_PI;
29.                         g3 = -sin((1-beta)*M_PI/(4*beta))*4*beta;
30.                         g4 = -2*M_PI;
31.                         h[n] = (g1+g2+g3)/g4;
32.                     } else {
33.                         t4 = 4*beta/(M_PI*sqrtf(T)*(1-(16*beta*beta*z*z)));
34.                         h[n] = t4*( t1 + (t2*t3) );
35.                     }
36.                 }
37.             }
38.         return h_len;
39.     }

```

Fig. 30. DesignRRCFilter Parallelized Algorithm Design


```

1. CalculateDerivativeFilterCoefficientsParallel ( )
2. {
3.     unsigned int h_len=8;
4.     unsigned int Npfb=8;
5.     unsigned int N = h_len*Npfb;
6.     float * dH = (float*)malloc (N * sizeof(float));
7.     float * H =(float*) malloc (h_len*sizeof(float));
8.     unsigned int i;
9.     for (i=id; i<N; i+=N_Processor)
10.    {
11.        if ( i==0 ) {
12.            dH[0] = H[1] - H[N-1];
13.        }
14.        else if ( i==N-1 ) {
15.            dH[N-1] = H[0] - H[N-2];
16.        }
17.        else {
18.            dH[i] = H[i+1] - H[i-1];
19.        }
20.    }
21.    free(dH);
22.    free(H);
23. }

```

Fig. 31. CalculateDerivativeFilterCoefficients Parallelized Algorithm Design.

These three algorithms are parallelized with the shown strategy using the designed parallelization primitives explained in Sub-section 7.1. The results of using these parallelization primitives on OSSIE Filter Functions using our designed parallelization strategy are shown in Table 7.

Filter Function	Single processor Time	Speed up with 16 processors
DesignRRCFilter	928,204	14.39
CalculateRRCFilterCoefficients	35,116,101	15.23
CalculateDerivativeFilterCoefficients	40320	6.56

Tab. 7. Speed up of OSSIE Filter Functions using Multi-processor platform

8. Additional Functionality in SDR waveform:

The reconfigurability offered by the general purpose platform defined in Section 4 makes it highly exploitable for all the SDR waveform components embedded implementation, irrespective of their algorithmic structure. In this Section, we partition the two SDR waveform resources: FFT and Viterbi

decoding on MPSoC. These two algorithms constitute the second class of SDR waveform components as they require considerable amount of intercommunication, while computing. These both algorithms are different from each other in the way they intercommunicate, using the designed data and synchronization NoC.

The parallelization efforts of FFT and Viterbi Decoding have been done by simulations [80,81] which do not take into account the memory synchronization. The automatic parallelization extraction tools for embedded architecture [82] penalize the performance by poor resource utilization. The specialized coprocessor approach [83, 84] is also in contrast to our proposal as we target a general purpose MPSoC. Since we target our efforts towards the SDR waveform reconfigurable components design, we feel that it is important to site the concept of Parameterization, which leads to a single procedure for all the common aspects of multiple Standards. This is accomplished by functions that are modifiable via parameter adjustment. There are two techniques of parameterization: top-down or Common Function (CF) [85] and bottom-up or Common Operator (CO) [86]. CO identifies the components, so as to reuse them throughout the waveform by adjusting their parameters. Palicot's CO Technique [86] demonstrated the replication of FFT in different transceiver components (Filter, Channel estimation, RAKE equalization, OFDM Modulation, Channelization), thereby making it exploitable by most of the functions of all the Standards. This notion is in contrast to CF Technique [85] where the components (Channel Coding, Modulation, Interleaving etc.) had to be Standards dependent in some way, by aggregation, for example. The CO can be determined by using any machine learning technique defined in Chapter 3. As an example, Simulated Annealing has been used to determine common operators in a multi-standard design scenario [87,88]. The Standards considered are WiFi (operating in three modes), WiMAX and UMTS. This Simulated Annealing based method is referred to as a Theoretical Approach in contrast to a two-fold Pragmatic Approach which is presented in [89], by Alaus. This pragmatic approach is applied to Reconfigurable Linear Feedback Shift Register (R-LFSR) and FFT, to find the common operators in Pseudo Random Sequences, Scrambling and Convolutional Encoder, in addition to components mentioned in [86]. These operations are dependent on a specific set of parameters [90]. Ghouwayel [91] investigates a 256 point FFT implemented on STRATIX-II, in a multistandard context, leading to another CO example. Naoues in [92] studies Viterbi Trellis and FFT Butterfly to decompose the two algorithms in different stages, eventually coming up with common operator architecture that reduces the complexity to 5% for both the algorithms under specific conditions. The Theoretical Approach for parameterization technique for CO method is further enhanced in [93] by Wang, where a radix-4 1024 point FFT architecture using R-LFSR based CORDIC operator is proposed. Thus, parameterization (using CO approach) is an important concept in the reconfigurable waveform component design, as it is an effort towards the same direction of multi-standard transceiver design, using reconfigurability. This work exploits the potential of actual NoC based MPSoC for FFT and Viterbi Decoding to add further functionality to

our designed SDR waveform. We target the inherent tedious intercommunication of the aforementioned algorithms and challenging memory synchronization tasks on the multicore that is implemented on a single chip.

8.1 FFT Parallelization Strategy:

Fast Fourier Transform is used for multi-resolution spectrum sensing in SDR applications. We have selected the radix-2 decimation in time (DIT) FFT algorithm as it is the easiest to exploit with respect to parallelization and thus implementation on our designed multiprocessor platform. Suppose p is the number of PEs over which we have parallelized and n is the number of transformed points of Discrete Fourier Transform (DFT). We have kept the value of n constant, 1024 and p is changed each time to get the number of clock cycles for different number of PEs.

The sequence of n points is divided into p consecutive subsequences, each of size n/p . Each subsequence is fed into individual PE, MicroBlaze. Our FFT algorithm consists of $\log(n) = 10$ steps, with each step to do butterfly computation with respective index difference. The algorithm is divided in two phases. For first phase (step 0 to step $\log(n/p)-1$), there is no communication between pair of processors. The master processor, responsible for execution of various control tasks in our multiprocessor environment, gives the launch signal to other processors via synchronized memory. The slave processors, upon receiving the master processor start signal, begin calculating their respective FFT coefficients in parallel and independently, until the end of first phase. Each slave

Each PE Computation for FFT	
1.	Input: $c = (c_0, c_1, \dots, c_{(\frac{n}{p})-1})$
2.	Output: $c = (c_0, c_1, \dots, c_{(\frac{n}{p})-1})$
3.	for $e=0$ to $\log(n/p)-1$
4.	{
	for each PE
5.	$l = 2^e, q = n/2l, z = w^q$
6.	for $(k=0$ to $n/p-1)$
7.	{
8.	if $(i*n/p+k) \bmod l = (i*n/p+k) \bmod 2l$
9.	{
10.	$m = (i*n/p+k) \bmod l$
11.	$c[k] = c[k] + c[k+l] * z^m$
12.	$c[k+l] = c[k] - c[k+l] * z^m$
13.	}
14.	}
15.	}

Fig. 32. Phase 1 of FFT Algorithm: Iterative Computation for each PE without communication

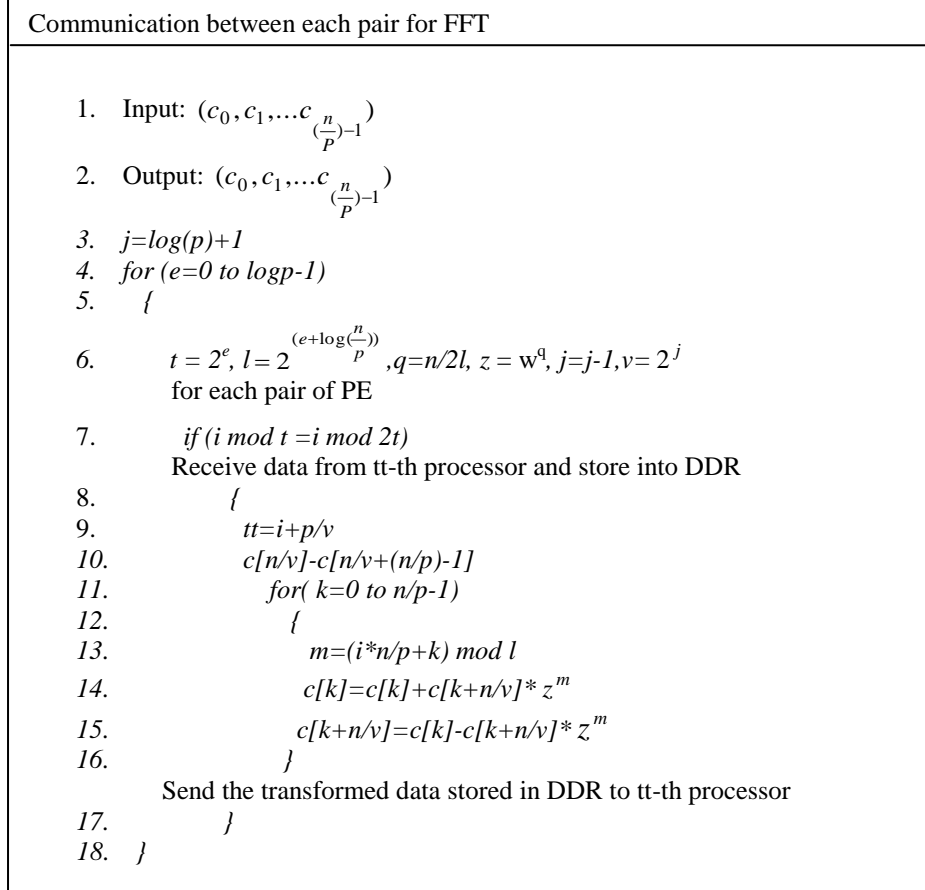


Fig. 33. Phase 2 of FFT algorithm: Communication between pairs of processors

writes a flag into the synchronized memory to signal that it has finished its job. During the second phase (step $\log(n/p)$ to step $\log(n-1)$), each processor communicates with another processor with the same communication primitives. At each step, in the second phase, there are $p/2$ processors computing in parallel after their communication with their corresponding other $p/2$ processors. Each processor computes $2n/p$ points. Communication between processors, being an overhead, occurs only during the last $\log(p)$ steps. At each of these steps, there is one data exchange between pairs of processors. Once the $\log(n-1)$ th step is finished by the pair of processors calculating the respective FFT points, the master PE captures the number of clock cycles. The two phases are shown in Figures 32 and 33, respectively.

8.2 Viterbi Decoding Parallelization Strategy:

Viterbi Decoding Algorithm for Convolution or Bose - Chaudhuri-Hocquenghem (BCH) codes is an error correcting code that allows the channel capacity to approach that of Shannon's limit, leading to efficient bandwidth utilization. It is frequently applied to maximum likelihood trellis decoding of linear codes, e.g. convolution codes. We treat the Viterbi algorithm in terms of the famous path cost

minimization problem that is closely related to matrix multiplication [94]. The algorithm is then parallelized by the row-wise partitioning of this matrix. $n - k$ bits are added to a message of k bits, so that each code word is n bits long. Suppose p is the number of PEs and the code parameters (n, k, d_{min}) are of the form, $n = 2^{m-1}$, $n - k \leq mt$, and the minimum Hamming distance is $d_{min} \leq 2t + 1$. The state of the Viterbi trellis is represented by a weight vector, \vec{w} , each element, w_s , of which gives the Hamming weight of the current path in state s . For the objective of decoding, it is easier to represent the state transition table as a matrix S in which the elements $s_{ij} = 1$, if state i can be arrived at from state j , otherwise $s_{ij} = 0$. Our parallelization strategy is to distribute row-wise the S matrix, resulting in a two processor solution if the matrix is cut in two halves, for each half the matrix vector reduction is performed independently of the other half, although each processor requires the entire weight vector.

The size or the number of states for code that we considered is 2^{16} for a code $(255, 239, 5)$. Channel output, $co[32p]$, is input to algorithm, whereas decoded output, $do[p]$, is the output where p is the number of processors. If the above explained dichotomy of the S matrix is continued, it will result in the generation of more complex task graphs, in which each processor represents a single state. Thus the parallel Viterbi decoding algorithm for a (n, k, d_{min}) code comprises of n matrix multiplications,

Each PE Computation for Viterbi Decoding	
1.	Input: $co[32p]$
2.	Output: $do[p]$
3.	for (state=0 to p)
4.	{
	for each PE
5.	prev_state0=(state<<1)
6.	$m0 = \text{partial_metrics}[\text{prev_state}0 \% p] + w[\text{ccodedot11_table}[\text{prev_state}0]]$
7.	prev_state1=(1+(state<<1))
8.	$m1 = \text{partial_metrics}[\text{prev_state}1 \% p] + w[\text{ccodedot11_table}[\text{prev_state}1]]$
9.	if ($m0 > m1$)
10.	{
11.	partial_metrics_new[state]= $m0$
12.	survivors[state][position]=prev_state0%p
13.	inputs[state][position]=(state>31)?1:0
14.	if($m0 > \text{max_metric}$)
15.	max_metric= $m0$
16.	}
17.	else
18.	{
19.	partial_metrics_new[state]= $m1$
20.	survivors[state][position]=prev_state1%p
21.	inputs[state][position]=(state>31)?1:0
22.	if ($m1 > \text{max_metric}$)
23.	max_metric= $m1$;
24.	}
25.	}

Fig. 34. Processing done by each PE for Viterbi Decoding Algorithm

each multiplication taking time proportional to n/p . After all processors have finished their assigned tasks, the total numbers of clock cycles are captured by master processor. The computation made by each PE is shown in Figure 34.

9. Experimental Results for FFT and Viterbi decoding Parallelization:

The number of clock cycles and speed up for each of these two algorithms is shown in Tables 8 and 9 respectively. Speed up is the ratio between number of clock cycles needed for computation on a single PE and number of clock cycles needed for computation on multiple PEs. Each of these two algorithms is ported four times changing the number of PEs each time to judge the effects of NoC characterized by bandwidth and latency. Each time we have configured our platform architecture to include 1, 4, 8 and 16 PEs respectively. The corresponding speed up for each of these configurations for both the algorithms as shown in Tables 8 and 9 is excellent. However, we observe that speed-up begins to diminish with the addition of more processors after the PE count is increased from eight. This is in contrast to the other algorithms (OSSIE Filter Functions) that we mapped on the same platform. In the case of FFT, the required communication overhead increases with the increase in the number of PEs because each processor communicates with its peer for synchronization. In the case of Viterbi decoding, the speed is solely the effect of communication cost between PEs to communicate the most feasible path. Also the task graph for Viterbi decoding changes with the number of PEs deployed to parallelize, and it tends to become more complex with the addition of processors more than 8. The speed-up for both the algorithms for different PEs is shown in Figure 35. We propose to further enhance performance efficiency by exploiting the fact that each PE, MicroBlaze, has the capability to execute its assigned tasks independent of the other fifteen PEs. We can divide the 16 PEs in further

Fast Fourier Transform		
Test Mode	Number of clock cycles	Speed up
Single processor	2,087,142	-
4 processors	668956	3.12
8 processors	348438	5.99
16 processors	187,693	11.12

Tab. 8. FFT Speed up for different number of PEs.

Viterbi Decoding		
Test Mode	Number of clock cycles	Speed up
Single processor	1667911	-
4 processors	483452	3.45
8 processors	277522	6.01
16 processors	145287	11.48

Tab. 9. Viterbi decoding Speed up for different number of PEs.

independent groups, e.g. in three groups of 8, 4 and 4 PEs, followed by the porting of FFT on the first group of 8 PEs and two Viterbi Decoding Algorithms, independent of each other, on the second and third group of 4 PEs, each. The total number of clock cycles, instead of being the sum of clock cycles for all three ported algorithms, 1,315,342, will be the clock cycles to execute the algorithm requiring the most number of clock cycles, 4,83,452 that correspond to Viterbi Decoding in this case. Thus the execution time is reduced to 63%. This means that we can even increase the number of PEs from 16 and can port other SDR resources such as Filter, Algebraic, Modulation and Demodulation Functions that constitute a complete SDR waveform. This way the complete SDR waveform can be mapped on the same chip exploiting the parallelism to address the area performance trade-offs for efficient SDR waveform embedded implementation.

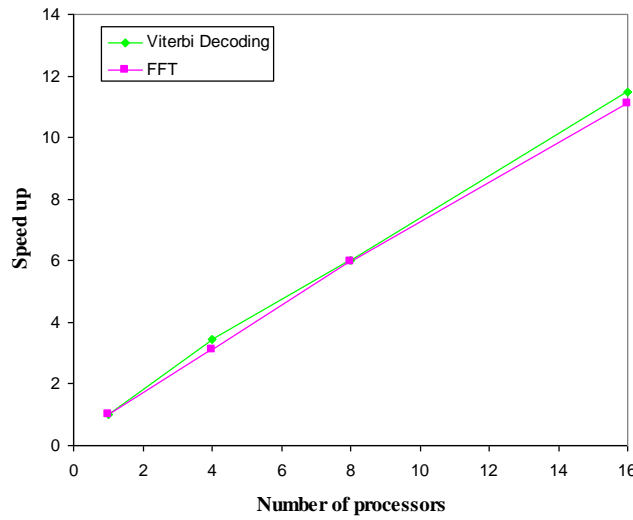


Fig. 35. Speed-up versus PEs for FFT and Viterbi Decoding

10. Other Functions to be realized:

We have addressed the parallel implementation of the most important functions and the other identified functions from OSSIE SigProc have been implemented on the embedded processor, Microblaze. There are many efforts that talk of more functionality in the baseband. In this regard, we have already cited the IDROMeL [37] project in Chapter 3. We have already defined the processing units: FFT, generic mapper for modulation detection, generic modulator, generic interleaver with rate matching, convolutional and cyclic encoder and viterbi decoder, addressed for two standards communication: UMTS and WiMAX. A single chip software defined IEEE 802.11.a and IEEE 802.11.b based communication task is achieved in [95], by implementing accelerators: FFT/IFFT, viterbi decoder, scrambler/descrambler, CRC, convolution encoder and FIR filter. The WINLAB network centric cognitive radio (WiNC2R) [96] is another CR platform whose baseband functionalities (FFT, viterbi decoding, Reed-Solomon coding) are implemented to deal with multiple

Standards. The dynamic switching between a number of OFDM and DSSS modems is the highlight of this effort. The Viterbi decoder portion of baseband functionality is implemented as ASIC in the project of Annabelle [97, 98], that deals with OFDM based standards. These efforts correspond to all the basic functions needed to implement this technology. We can compare these efforts to microprocessor development that started with the basic instruction processing general purpose computation engine, that we call microprocessor today, which has the capability to solve all the problems in the framework of the basic instructions: LOAD, STORE, IF JUMPS and CALL functions. The overall baseband functionality (at receiver side of the transceiver) that has to be software defined consists of a number of steps. The signal from antenna is selected for the frequency regions of the specific Standard by means of a preselection filter. This is followed by amplification, band limitation, A/D conversion, low pass filtering. The base band portion takes care of equalization, demodulation, decoding and despreading, if necessary. The equalization, that addresses the multi-path impairment in wireless channels, has been shown to be achieved by FFT in [86]. The other method to implement an equalizer is via finite impulse response (FIR) filter. Before the baseband processing, the Inphase and Quadrature components are generated. This complex mixing and filtering can be merged as addressed in [93]. In the context of a Multi-standard Transceiver design components, if we assume a transceiver design that takes into account the features of all the existing Standards: allocated frequency, signal power level, the essential signal to noise ratio, then it does not hinder in the addition of future expected standards, as the functionality remains the same with the adjustment of parameters. However, the differences and similarities between different Standards should be carefully explored. We can quote the example of Common Operators Parameterization approach that we have cited in Section 8 of this Chapter. The transmission functions are the counterpart of the functions that we have described so far, i.e. channel coding, modulation, spreading (in the case of CDMA, for example) and pulse forming. We must note that pulse shaping is usually a root-raised cosine rolloff filter addressed in equation (4.2) of this Chapter. The Inphase and Quadrature components are D/A converted and filtered by a filter tunable to different Standards. Before transmission, the signal is amplified and band pass filtered. As we have mentioned the exploration efforts of similarities between different Standard functions, we feel that it is important to illustrate it with an example. If we consider GSM, then the baseband function of channel coding is accomplished by convolutional encoder and interleaver. If we implement the convolutional encoder in such a way that it can adapt to perform a recursive systematic encoding, with multiple iterations, we can change the functionality from convolutional to turbo coding, which are a special kind of convolutional codes. Another modern type of channel coding is based on Reed Solomon (RS) codes that are a type of cyclic codes, known to correct burst-type errors in wireless and mobile communication units and satellite links. These promising codes are adopted by many wireless Standards, like WiFi and WiMAX. The Turbo decoding of UMTS is addressed in [99]. The third generation mobile systems use both: convolutional and turbo coding. The convolutional codes have proved to be a better choice for speech transmission, whereas turbo codes are employed to permit data

rates up to 2 Mbit/s in the case of bit error rates of less than 10^{-6} for particular applications. We also observe that the same signal processing function may be implemented either in same way or different way for different Standards, e.g. a trellis-based equalization scheme is implemented for GSM, whereas RAKE receiver for UMTS in [100]. At the same time, both Standards use the same channel estimation scheme. The baseband functions of different Standards: channel encoding, modulation, spreading, have been studied in detail to find the similarities between the same functional block for different Standards [101]. Source encoding and encryption are important components of a communication chain. We have addressed the FFT implementation and have already cited its use as a CO in Section 8, which is presented in [86, 91]. The tasks include filter functioning, channelization, channel estimation, correlation and spreading/despreading. The authors [89] present the idea of different blocks: pseudo random sequence generators, scramblers, convolutional coder, cyclic redundancy check and block channel coding, implemented by Reconfigurable Linear Feedback Shift Register. Furthermore, the functions of scrambling, CRC and convolutional coding are replaced by different LFSR operators for different Standards. Delahaye [102] groups the baseband functions comprising a multi-standard functionality into three functional classes: coding, data handling and modulation. The coding class includes functions like cyclic coding, convolutional and turbo coding. Data handling class manipulates data packets by concatenation, segmentation and multiplexing functions. Modulation class includes RRC Filtering, mapping, scrambling and spreading.

Thus, in this Section, we gave a brief overview of the other building blocks needed to realize an efficient Multi-standard Transceiver. Although we have addressed the most important functions needed by all the Standards, the parallelization of many building blocks to map them efficiently on the addressed MPSOC, discussed in this Section remain a future perspective to add further functionality, with respect to the treatment of all the Standards, in the designed waveform.

11. Conclusions:

This chapter showed that significant progress has been made towards making portable software radios commercially viable. We described in detail our efforts that contribute to this endeavour. We described our target 16 Processing Element Network on chip based general purpose Multiprocessors System on chip (MPSoC), implemented on a single chip Xilinx Virtex-4 FPGA. This general purpose MPSoC is used to port the SDR waveform components after designing an efficient parallel implementation strategy. We started with the Signal Processing Library of OSSIE that implements DSP algorithms. We studied SigProc in detail and identified the functions requiring optimization and then implemented them in an optimized fashion after deciding the appropriate cache size needed for each function. We parallelized the optimization requiring functions and ported them on our MPSoC, gaining significant speed up. We moved one step further by optimized parallel implementation of two SDR resources, FFT and Viterbi Decoding, after designing an efficient parallelization strategy on the

same platform and obtained excellent speed-up. We showed that we can benefit from the capabilities of multiprocessor platforms to enhance the performance of computationally intensive SDR waveform algorithms while keeping the flexibility that such platforms allow, in particular by shifting short term development into software domain. However we are reminded of the limitations and problems that can occur when adding more processors. We addressed this limitation by partitioning the PEs for different waveform components and thus achieving optimized platform utilization. In the last Section, for the sake of completion, we give a list of functions that have been implemented by researchers – and not parallelized by us – to realize Multi-Standard Transceiver design.

The next Chapter sheds light on our efforts to enhance our designed SDR waveform by cognition incorporation, so as to change it into a cognitive waveform. We have already introduced the idea of a Universal Transceiver capable of operating in different Standards mode and switching to the appropriate configuration, in Chapter 3. In this framework, the next Chapter explains our strategy to engineer such a Multi-Standard Universal transceiver using machine learning technique of Artificial Neural Networks. It addresses our motivation to choose this very machine learning technique amongst the available techniques that we described in Chapter 3.

Chapter 5:

Standard Recognizing Artificial Neural Networks Based Reconfigurable Cognitive Radio Transceiver and its Embedded Implementation.

1. Introduction:

We have already defined a Cognitive Radio as a radio that has awareness of changes in its environment and in response to these changes adapts its operating characteristics in some way to improve its performance or to minimize loss in performance. This adaptation of characteristics in *some way* is where the Artificial Intelligence (AI) techniques come into play. Being aware of the existing operators in the environment, CR should choose the best available option based on performance for each application. The potential for cognition incorporation in a radio has also been found by IEEE that can be witnessed by IEEE 802.22 protocol specifications that aim at design of a new cognitive radio interface. The choice of decision algorithm to adapt the radio characteristics accordingly is a challenging topic, since several optimization schemes are available in literature. We will analyse all these strategies in this chapter, justifying our choice of Neural Networks Selection.

Cognitive Radio Systems have transceivers with the ability to adjust their operating parameters after observing the results in order to decide to operate in a specific radio configuration, expecting to move the radio towards some optimized operational state. Furthermore, CR system design has to be based on single chip multiprocessors as the mode switching from one radio configuration to another is a real time constraint and only single chip multiprocessors address such high performance and flexibility requirements. Framed within these two statements, this chapter proposes and implements three Neural Networks (NN) Schemes: Self Organizing Maps (SOM), Linear Vector Quantization (LVQ) and Multi-Layer Perceptrons (MLP) as effective techniques for reconfiguring transceivers after

recognizing the specific standard based on input parameters extracted from the signal. We exploit the inherent property of SOM, Tonotopy [107], to recognize the standard in our proposed multi-standard cognitive transceiver. In the case of MLP, we have used Mean Square Error (MSE) as a metric for measuring the MLP's performance. We implement our proposed multi-standard CR Transceiver based on aforementioned three Neural Networks, on 16 Processing Element (PE) Network on chip (NoC) based general purpose Multiprocessors System on chip (MPSoC), implemented on a Xilinx Virtex-4 FPGA, in order to meet the challenge of the ITRS Roadmap prediction [60] that directly limits the capabilities of agile frequency operations of CR. Each MPSoC resource has different execution frequency according to its requirement. We obtained speed-up greater than 13 for all three algorithms, designed to recognize the standard based on received signal parameters, that verifies our approach of Neural Networks based Cognitive Transceiver implementation on general purpose MPSoC. We target this work at cognition incorporation in our designed Software Defined Radio (SDR) waveform.

Multimode reconfigurable devices adapt their behaviour to the changes in environment. As an example, second and third generation cellular radio access technologies and 802 wireless standards can be quoted. In this respect, future CR devices, candidates for 4G radio interface networks, will have the capability to optimize their state thereby adjusting their operating parameters accordingly, after sensing the environment variables. This is explained in cognition cycle [23]. One such example of moving the radio in an optimized state after sensing the parameters is that of an universal transceiver that is capable of operating in many transmission systems and that can be made to operate in a desired standard, by intelligence incorporation. In this work, we deal with such intelligence incorporation techniques in the transceiver. We address the problem of intelligent detection of a wireless standard among a predefined list of standards to address the challenge of multimode reconfigurable devices. We target at the identification of characteristics of a signal, pertaining to different layers of a communication system, thus cross-optimizing cognitive transceiver. We propose to identify the parameters specific to a particular standard using Unsupervised Neural Networks called Self-Organizing Maps (SOM) and Linear Vector Quantization (LVQ) [104] and Supervised Neural Neural Technique, MultiLayer Perceptron (MLP) [105]. The difference between Supervised and Unsupervised NNs is the existence or non-existence of desired output during learning. Multi-Layer Feed Forward NN (MFNN) and Multi-Layer Perceptron are the most common examples of Supervised NN. Unsupervised NN Techniques are more suitable in building CRs that require minimal pre-configuration. MLP, being a Supervised Neural Network, is exploited to have an estimate of transmission power needed in the case of huge data range, as in the case of IEEE 802.22 cognition incorporation standard [106]. We explore in detail the decision between the three Neural Networks, based on signal features detected in Section 4. The motivation to use SOM and LVQ for our standard recognition problem comes from their inherent property of Tonotopy [107] explained in our problem context in Section 4.2.

Another reason to choose LVQ, SOM and MLP based cognition incorporation algorithms is their inherent tendency towards parallelism that facilitates to address the gap between software and hardware design productivity. Since future generation System on chip (SoC) will experience an exponential increase in the number of processing engines and processors, we have exploited the inherent Neural Network parallelism in the framework of CR embedded implementation on our multi-core chip. Also MPSoC have many advantages, such as reduced short term deployment cycles because of which newer versions and new functionality, such as a new parameter to be detected in the received signal by CR, can be added through software development. This makes MPSoC, together with NN Techniques, a strong candidate for efficient CR multi-standard transceiver embedded implementation. There has been some proposals about reconfigurable radio platform, e.g. Q.Zhang [97, 98] proposed a heterogeneous System-on-chip consisting of General Purpose Processor (GPP), Domain Specific Reconfigurable Hardware (DSRH), DSP, FPGA and ASIC tiles, connected by router based NoC. However, the basic limitation lies in unused tiles, when an application is mapped on appropriate tiles. This limitation results in inefficient resource utilization. We overcome this limitation by an efficient load balancing across all 16 Processing Elements (PE). The speed-up we obtained for our cognitive NN algorithms demonstrate LVQ, SOM and MLP as potential solution to the standard recognizing multi-standard CR transceiver embedded implementation. In the next section, we go through a brief review of Neural Networks and their applications.

2. Neural Networks Review:

Artificial Neural Networks (ANN or simply NN) are made up of artificial neurons interconnected to each other to form a programming structure that mimics the behaviour and neural processing (organization and learning) of biological neurons in human brain. NNs try to mimic the biological neurons to solve the problems with cognitive or associative tinge. It is because of this reason that neural networks have extensive and huge applications. They are known to have applied successfully to speech recognition, image processing, time series prediction, function approximation, classification, recognition, adaptive control and many other areas. The architecture of NNs has three types of computing units or neurons: input neurons that deal with the external stimuli by accepting the data to be processed from the external environment, which are organized in the so called input layer, output neurons that send the data to the external environment comprising of output layer, and hidden neurons which are the processing units composed of one or more hidden layer(s). The input and output signal of hidden layer (layers) neurons remains inside the network.

Irrespective of any application, a NN is required to be configured in such a way that a specific set of inputs from any application gives the desired output. This task is accomplished by proper adjustment of connections among neurons of different layers. This connection between any two neurons of

different layers is called weight. The adjustment of weights in such a way that a known input produces the desired output is called learning. To this effect, NNs work in two modes: Learning and Testing. Before putting a NN into operational phase, it is required to learn about the features of a specific application. This learning task can be of two types: Supervised and Unsupervised. Supervised Learning is applied in cases when the desired output is known well ahead in time. During learning phase, the NN is given with the teaching patterns of input so that it modifies its weights to converge to the desired output according to some learning algorithm. This weight modification leads to the match of actual output with the desired known output. The Unsupervised Learning is applied in the cases where the desired output is not known in advance. The NN, in Unsupervised Learning, makes use of statistical techniques to learn about the remarkable features of the applied input patterns. This way, NN classifies itself to adjust its weights accordingly till the right output iteration is achieved. The way NN operates is similar to human brain operations. The general principle of connection of neurons is also inspired by human brain in such a way that the transmission activity of a group of neurons to another group of neurons via connections, transfers the information to reach the correct output. There are many models and topologies of Artificial Neural Networks, inspired from the rules and principles of biological neurons. All topologies provide the following functions:

- Knowledge is acquired through a process of training.
- Each connection is provided with an adaptive weight.
- The training is done by modification of connections.

The neuronal approach is opposed to the symbolic system approach that is based on assumption where the reasoning modelling is a combination of symbols being subject to the logical rules. Thus Neural Network approach has the privilege of the following advantages.

- Parallel real time activity.
- The distributed representation of knowledge.
- The training by modification of connections.

However, there are certain limitations of Artificial Neural Networks, such as,

- There is no systematic rule to determine the network architecture.
- There is no analytical solution so as to where to use the iteration.
- They take rather long time for training and optimization.
- Various initial conditions of the weight values can lead to various solutions that have different performances.
- Difficulty of interpretation without expert testimonies.

We can overcome these limitations if we select the appropriate learning method followed by appropriate algorithm. Normally, a supervised learning algorithm is known to overcome these inconveniences. However, this is not a hard and fast rule. There are many applications which are inherently suitable for unsupervised learning. The proper readjustment of the probability distribution at the output side will make it possible to adapt the NN according to our application needs. We describe the structure and architecture of three neural networks LVQ, SOM and MLP that we have used in this chapter.

2.1 Structure and Architecture of Artificial Neural Networks: LVQ, SOM and MLP:

In this Chapter we have used three neural network models: LVQ, SOM and MLP. The majority of these classifiers consist of multi-layer networks of neurons. These kind of networks have the advantage of solving the problems of data classification, non-linearly separable. In SOM, the neurons can learn how to classify input vectors in a way that the same group of neurons are grouped in a space. This differs from other neural networks where the neurons and their neighbors learn how to recognize grouping within the entry space. There is a layer of competitive neurons that consists of an input vector with N elements connected by weights to m neurons. The output of competition layer consists of Euclidean distance calculated for each neuron. The neuron having the minimal distance is declared to be the winning neuron. For the training, the rule of evolution of the weights of the neurons follows the Kohonen except that instead of activating the single winning neuron (minimal distance), all neurons being in vicinity (lower than a distance D , for example), will also be activated. The mathematical rules for update and neighbourhood function will be explored further, later in this chapter.

The architecture of competitive layer for SOM and LVQ is made up of two layers: an input layer with N entries and an output layer made up of $k=p*q$ neurons. This enables us to present the second layer known as the class separation for the SOM and classification for the LVQ as being common to both Neural Networks. The characteristics of model that belong to this category lies in the fact that relevant information will be extracted only from the organization of the data presented to the input.

The MLP is characterized by the presence of one or several hidden layers, whose corresponding calculation nodes are called hidden neurons or hidden units. The hidden layer interposes between the entry and exit of the network. The role of hidden layer is to carry out a pre-treatment signal received by the input layer coming from the external medium, and to transmit the results corresponding to the output layer where the final output of the network will be given, before it is transmitted to the external medium. The entries of the neurons of a particular layer come from the exit of the preceding adjacent

layer. The activation function used for the neurons is derivable. The functions used generally are threshold, semi-linear or sigmoid. We will delve in detail of our used activation function later in this chapter.

3. Related Work:

There have been some proposals describing cognition incorporation within a radio. Traditional CR proposals rely on a priori characterization of the radio states, which are often derived from analytical models, e.g. in [109] analytical models of bit error rate (BER) performance of different schemes are used to derive three objective functions, BER, throughput and power consumption. These objectives are then evaluated in the process of optimizing the chosen physical (PHY) layer configuration. Van [110] describes a generic framework for cross layer optimization of multimedia communications in which analytical models are used to define modulation and channel coding schemes for PHY layer, different packetization, automatic repeat request (ARQ), scheduling and forward error correction (FEC) mechanisms for MAC layer and then propose cross layer optimization strategy for these objective functions of different layers. These analytical models are not always practical due to the limitation of modelling assumptions and non-ideal behaviour in real life scenarios. In order to overcome this limitation, there are more pragmatic approaches, that talk of Genetic Algorithms [111, 112] and Fuzzy Logic [113]. However, the common drawback of both the approaches is that they do not provide any means of learning from past experiences, thus failing to exhibit one of the key properties of CR. Also the convergence in Multi-Objective Genetic Algorithms is slower and that makes this solution less viable in real time scenario of CR parameter detection [114]. In [115] Markov-based Channel Prediction Algorithm (MCPA) is presented for dynamic spectrum allocation in cognitive radio networks. However Hidden Markov Models, in general, can not be considered for performance modelling in CRs, primarily due to the difficulty of representing the type of input/output relationship needed for orientation, as explained in cognition cycle. Another proposal talks of Swarm Intelligence [116]. All these CR proposals have a common drawback, i.e. if any new parameter has to be added in the system input, these models have to be re-designed from scratch.

The motivation towards neural network based solutions to the standard recognition problem in CR scenario comes from all these observations. Cognitive Radio can effectively learn as we train the neural network characterizing system performance with data obtained from observations performed by the CR itself. There has been some work based on intelligence incorporation using neural networks in CR. Zhang [117] simulates an Error Back propagation (BP) Neural Network in MATLAB that considers changeable and unchangeable parameters of Cognitive Radio. A.F. Cattoni [118] presents Mode Identification and Spectrum Monitoring (MISM) process that uses the features extracted from incoming radio signal, which are fed to a three layered Neural Network that uses MSE on training set as a cost function. Clancy [119] simulates SOM for false signal detection of primary user based on

feature detection. He used MATLAB Neural Network Tool Box, that, unlike our choice, SOM Tool Box does not provide with all the features of SOM. Tsagkaris [120] uses a set of 3000 training data input values to predict the anticipated capabilities taking into account recent information sensed. However, his work is oriented towards data rate prediction only, rather than standard identification. Baldo [121,122] presents MFNN to predict the throughput performance of an infrastructured 802.11 cell, leading to dynamic channel selection. Ustundab [123, 124] uses MATLAB to simulate MLP Neural network based decoding receiver, which uses frequency and amplitude to distinguish between signal patterns leading to an efficient communication bandwidth. MLP and MFNN are supervised learning schemes that make them less pragmatic for our application as we want to recognize the transmission system in use without any prior knowledge about the signal at the receiver, i.e. desired outputs do not exist during learning procedure. Bixio [125] distinguishes between two transmission standards based on their extracted features using Simplified Fuzzy ARTMAP (SFAM) neural network ensembles. In a similar effort to ours, Hachemani [126] proposes the idea of a standard recognizing sensor, although his proposal is limited to physical layer features pertaining to a standard. The implementation details of such a sensor are not addressed. For each standard to be recognized the features to be tested are different, thus leading to some a priori knowledge regarding the standard to be recognized.

This overview shows that there has not been any effort in considering the parameters pertaining to different layers to recognize a standard. We consider application layer parameters such as localization and positioning characteristics as well, a part from physical layer and transport layer features of a signal that makes our work unique. Our work is also different from all these proposed approaches as it uses an adaptable solution that exploits two unsupervised NNs and one supervised NN to recognize different standards and in the case of huge data range, the estimated transmission power level required, thereby allowing CR to switch in the specific standard mode, optimizing the radio configuration to give the best Quality of Service (QoS) and self management capability. Our solution is adaptive as it switches to appropriate Neural Network either, SOM, LVQ or MLP based on the features of input signal sensed. We can also add further more parameters from different layers to increase the performance of our CR.

4. Our proposed Approach:

We analyse the received signal which may be the result of the summation of many standards, each standard itself the summation of several modulated carriers. Analysis of this received signal provides important information needed by the receiver. The analysis of the received signal to extract features is followed by the choice between LVQ, SOM and MLP. This choice is decided upon by these extracted

features while analysing the received signal. Hence in this work, we have used three different data sets, each corresponding to the selection of a unique Neural Network. These data sets are actually the detected features of our received signal. This scenario is visualized in Figure 36.

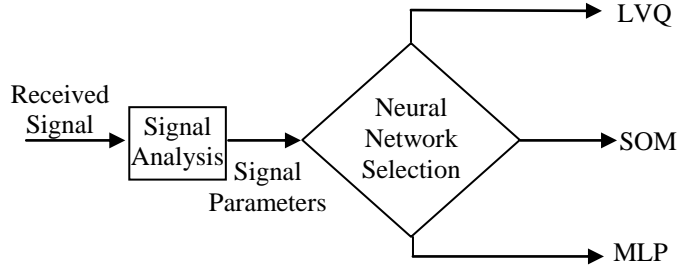


Fig. 36. Neural Networks Selection based on received signal extracted parameters.

We describe in detail the scenario shown in Figure 36 in Sub-section 4.1. Subsection 4.2 explains our implementation of LVQ and SOM Unsupervised NN in our problem context. We explain our MLP Supervised NN implementation in Sub-section 4.3 and we conclude this Section by presenting and explaining our Standard recognition experimental results in Sub-section 4.4.

4.1 SOM, LVQ and MLP for CR Standard Identification:

The predefined list of Standards to be recognized is divided in three categories as shown in Table 10. The choice between three Neural Networks: SOM, LVQ and MLP is decided upon by the detected features of the received signal. Then the identified features are fed as input to the identified trained NN, resulting in the identification of Standard, and in the case of huge data rate, the identification of transmission power needed as well. After recognizing the Standard, the receiver switches to that specific standard mode to detect the right demodulation software or download the appropriate software. The first category comprises of fifteen Standards. Each Standard in this category is recognized on the basis of one single parameter: channel bandwidth, shown in Table 11. As LVQ follows *winner takes all* strategy, it is the natural solution for standard recognition for this category of standards. An earlier effort to recognize these 15 standards attempted the use of Supervised Radial Basis Function Neural Network [127]. We propose LVQ because of their easy adaptability to SOM to recognize a number of more Standards by just removing neighborhood function. We explore our used neighborhood function in Sub-section 4.2. The second category of ten standards with the identification parameters is shown in Table 12. These Standards are identified by the help of four parameters: channel bandwidth, frequency band, data range and data rate. Every Standard needs a neighborhood function, as it is probable that the next signal may be of another closely related standard making SOM a natural choice in this case. We have chosen LVQ and SOM for Table 11 and Table 12 standards as they inherently suit our problem, because of their property of Tonotopy (equation (5.5)). The third category comprises of 802.22 Standard (802.22.1 and 802.22.2) being developed for Wireless

Regional Area Network (WRAN) using white spaces in the TV frequency spectrum. The data range is huge in this category of standards and thus we need the estimated transmission power level required. As shown in Table 13, 802.22 requires the power level of 4 watts to achieve the data range of 33 km at the channel bandwidth of 6 MHz, whereas the data rate is 18-24 Kbps. In such a scenario when we know the data rate and bandwidth and are aware of huge range, we are in need of power level estimation. In this work, the data range greater than 30 km is defined as huge data range and the Standard available to address the huge data range requirements in our predefined list of standards is 802.22, whose transmission power needs to be known in advance and hence the supervised NN, MLP, justifies our need to estimate the desired power level. In the NN Selection phase illustrated in Figure

Candidate Standards for LVQ	Candidate Standards for SOM	Candidate Standard for MLP
PDC	802.11.g	802.22
ADC (D-AMPS)	802.15.4	
	802.11.b	
CT2	802.15.1	
GSM	802.11.a	
PHS	802.11.n	
Bluetooth	802.11.h	
IS95, Global Star	802.11.j	
	802.16.d	
DAB	802.16.e	
DECT		
UMTS (FDD)		
DVB-T, LMDS		
Hiperlan 1		
DVB-S		
LMDS		
Hiperlan2		

Tab. 10. Standards identified by respective Neural Network Techniques.

Standards	Channel Bandwidth
PDC	25 kHz
ADC (D-AMPS)	30 kHz
CT2	100 kHz
GSM	200 kHz
PHS	300 kHz
Bluetooth	1 MHz
IS95, Global Star	1.25 MHz
DAB	1.712 MHz
DECT	1.728 MHz
UMTS (FDD)	5 MHz
DVB-T, LMDS	7-8 MHz
Hiperlan 1	20 MHz
DVB-S	32-36 MHz
LMDS	32-36 MHz
Hiperlan 2	50 MHz

Tab. 11. Parameters of Standards recognized using LVQ

36, if the data range is huge enough to require considerable amount of transmission power, the selected NN is MLP, and hence the estimated transmission power for the detected data range has to be matched with target transmission power.

As shown in Figure 37, these distinguishing parameters for each standard are the inputs to the first state of cognition cycle. This information is passed to the Configuration Selection state of the cycle and this is where our proposed Neural Network based solution comes into play. We focus on the data Configuration Selection phase of the cognition cycle using LVQ, SOM and MLP. We construct the data by taking samples of each Standard. Each Standard corresponds to 50 signals. Each row represents specific Standard and each column represents a specific parameter of all signals of all

Standards	Channel Bandwidth	Frequency Band	Range	Data Rate
IEEE 802.11g	20 MHz	2.4 GHz	140m	54Mbps
IEEE 802.15.4	2 MHz	915 MHz	75m	40 kbps or 250 kbps
IEEE 802.11b	1 or 25 MHz	2.4 GHz	100m	11 Mbps
IEEE 802.15.1	1 MHz	2.4 GHz	10 m	3Mbps
IEEE 802.11 a	20 MHz	5.4 GHz	120m	54Mbps
IEEE 802.11n	20 MHz	2.4/5GHz	250m	248Mbps
IEEE 802.11h	20 MHz	5 GHz	100m	-
IEEE 802.11j	20 MHz	2.4 GHz	120m	54Mbps
IEEE 802.16d	20 MHz	2 to 11 GHz	7 km	75 Mbps
IEEE 802.16e	20 MHz	2 to 6 GHz	3.5 km	30 Mbps

Tab. 12. Parameters of Standards recognised using SOM

Standard IEEE 802.22 Identification parameters:			
Power Level	Data Range	Channel bandwidth	Data rate
4 watts	33 km	6 MHz	18-24 Kbps

Tab. 13. Parameters of Standard recognized using MLP

standards. Then we normalize this data so that the data range of each parameter is given equal importance, e.g. if the channel bandwidth is 20 MHz and data rate is 2 Mbps, then without normalization, the channel bandwidth will almost completely dominate the output grid, which is not desirable. We avoid this by normalizing the variance of all the parameters to unity and its mean to zero, thus giving all parameters equal weight in Euclidean distance calculation. Then we first choose the output grid size, based on our data, followed by initializing the map using linear initialization and finally we use batch algorithm to train the map. The batch algorithm uses all the standard samples to present it to the map before any adjustments are made. Once all samples are presented to SOM, our proposed iterative structure of the algorithm starts calculating the Best Matching Unit (BMU), using the designed algorithm we define in next Sub-sections. Each map unit has two sets of co-ordinates, the prototype vectors in the input space and position on the map in the output space. In the case of SOM,

the training is done in two phases: first with large neighbourhood radius, and then fine tune with small radius. By using a large neighbourhood radius, the property of Tonotopy is exploited. The implementation of each of these NN is explained in Subsections 4.2 and 4.3.

4.2 SOM and LVQ Implementation:

We have used the MATLAB SOM toolbox [128] to implement the SOM and LVQ explained in this section. Each row in the MATLAB 'struct' represents specific standard and each column represents a specific parameter of all signals of all standards. We have named the 'struct' as *Standard*. The labels

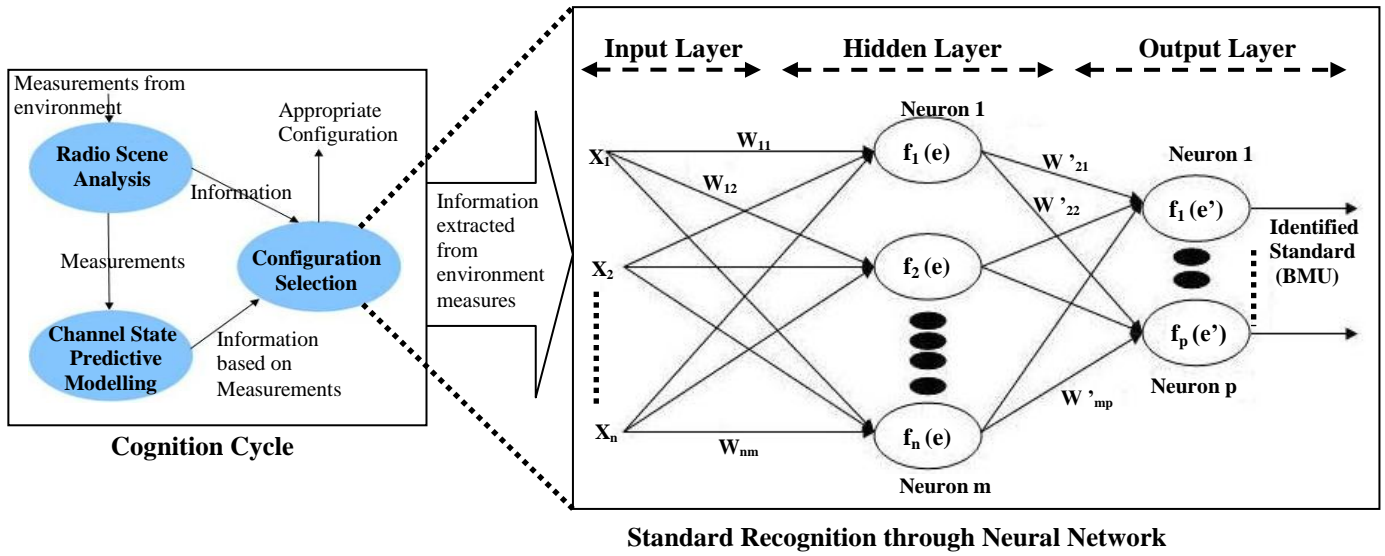


Fig. 37. SOM and LVQ and MLP based Cognition Cycle Receiver.

field represent the specific standard that is formed by the combination of all four variables. SOM has properties of vector quantization that makes it suitable for easy LVQ adaptation with minor modifications. If prototype vectors are positioned on a regular grid in an ordered fashion, three execution phases are to be considered. The first phase consists of computing the distances $d_j(j=1,2,..k)$ between the input vectors and weights of various neurons using Minkowski distance function (L_2 norm) shown in equation (5.1).

$$d_j = \sqrt[p]{\sum_{i=1}^n (X_i - W_{ij})^p} \quad (5.1)$$

$X \in R^n$ where n is the number of network's input features and $W \in R^{nk}$, where k is the number of neurons. For $p=2$, it represents Euclidean distance function which is referred to as L_2 norm and gives better precision than L_1 norm (Manhattan distance function). During the second phase, the winning neuron called Best Matching Unit (BMU) is determined by comparing these distances. In the third

phase, the weight of BMU, W_c is updated using Subtractive method of weight updates, using equation (5.2).

$$W_j(t+1) = W_{ij}(t) + h_{ci}(t)[X_i(t) - W_{ij}(t)] \quad (5.2)$$

If LVQ is used, then only winning neuron's weight is updated, which means that there does not exist any neighborhood function. However, in the case of SOM, weights of neighboring neurons are updated as well. The neighborhood defines the region of influence that the input sample has on SOM. This concept is related to the inherent property of SOM, Tonotopy, which is the special arrangement of close data sets extracted from the signal. The neighborhood function around the BMU is Gaussian and is defined by $h_{ci}(t)$ which is a non-increasing function around the BMU and is given by equation (5.3), where r_c and r_i are the distances of winning neuron and neighboring candidate neuron respectively, $\sigma(t)$ is the neighborhood radius at time t and $\alpha(t) \in [0,1]$ is the leaning rate for each epoch and is a decreasing function of time. The neighborhood radius used is $\sigma(t)=2$.

$$h_{ci}(t) = \alpha(t) \exp \left[-\frac{\|r_c - r_i\|^2}{2\sigma^2(t)} \right] \quad (5.3)$$

The neighborhood shown in above equation is a non-increasing function of time and of the distance i , from the winner neuron c . It is because of this neighborhood, that after finding the BMU, not only BMU's weight is updated, but also the weight vectors of the neighbours are moved closer to the input vector. The whole procedure of the algorithm is depicted in Figure 38.

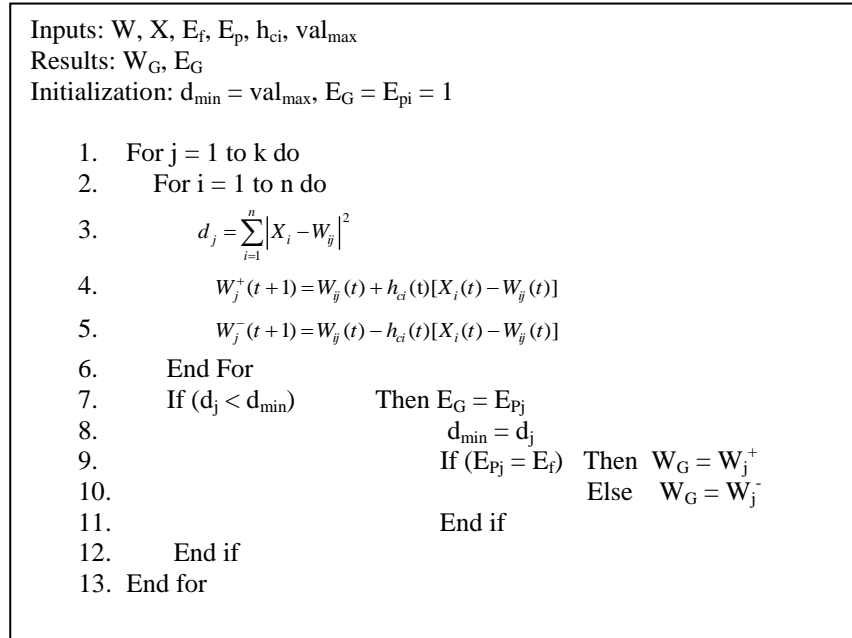


Fig. 38. SOM Algorithm for BMU (Standard) determination and weight update.

The description of the algorithm shown in Figure 38 is summarized in the following steps:

1. Initialization: the iteration number T , and the initial weights matrix W . let $t=0$.
2. For each input vector X and current W :
 - i. Select the nearest prototype to the input vector X

$$C = \arg_j \min \left\{ \|X - W_j\| \right\}, j = 1, 2, \dots, k \quad (5.4)$$

- ii. Calculate the updating weights W_c vectors

- $W_c(t+1) = W_c(t) + \alpha(t)[X(t) - W_c(t)]$ if class $X = \text{class } W_c$
- $W_c(t+1) = W_c(t) - \alpha(t)[X(t) - W_c(t)]$ if class $X \neq \text{class } W_c$.

3. If $t < T$, $t=t+1$, go to step 2.

4. Output updated W .

The above summarized four steps are meant for LVQ as only the winning neuron's weight is updated, which means that there does not exist any neighborhood function. However, in the case of SOM, weights of neighboring neurons are updated as well.

Inputs to algorithm are:

- W (a matrix with $n \times k$ elements).
- X (an input vector with n elements).
- E_f (reference label).
- E_p (k expert's label).
- h_{ci} (neighborhood function) for SOM, or α (learning parameter) for LVQ, depending on extracted features of signal.

The algorithm gives the following results:

- W_G (weight of the winning neuron).
- E_G (label of the winning neuron).

The following two principles are followed while calculating new weight vectors:

1. **Competitive Learning:** The proto-type vector most similar to a data vector is modified so that it is even more similar to it, called positive weight calculation. This way map learns the position of the data cloud.

2. **Co-operative Learning:** Apart from the most similar prototype vector, its neighbours are also moved towards the data vector. This way the map self-organizes.

Also, the negative weights are calculated to increase the distance between input vector and weight, in case of least similar vectors. In the case of LVQ, h_{ci} is replaced with α . In Figure 38, after initialization, the algorithm calculates in two overlapping loops (loop 1, line 1– 13 and loop 2, line 2 – 6) the distances, the positive and negative weights, weight update of winning neuron and its label assignment.

We have chosen SOM and LVQ as they inherently suit our problem, because of their property of Tonotopy. Since our proposed transceiver recognizes the specific Standard based on the clusters at the output grid, it is our inherent requirement that any two spatially close data sets with respect to their parameters should be mapped onto two topographically close cells. We achieve this requirement by this very property of Tonotopy. The word Tonotopy is derived from Greek word “Tono” , which means space. Kohonen illustrates how self organizing maps approximate various vector sets in spatial orderly fashion [104]. He shows that because of this property, the prototype vectors are positioned on a regular low dimensional grid in an ordered fashion. Furthermore, this property has been exploited by many researchers for grouping and visualizing different data sets in different applications [107]. This property states, in our problem context, that the spatial location of a specific standard corresponds to a particular feature of input patterns, so that all the signals from a particular standard are plotted at the same place at the output grid, thus forming a cluster. Mathematically, let $P(a,b,c,d)$ be the four distinguishing parameters, namely channel bandwidth, frequency band, range and data rate considered for a signal. We quantize this set of four parameters using a 2-D map, M , organized in a hexagonal grid, so the mapping K is defined as,

$$K : P \rightarrow M, p = (a,b,c,d) \in P \rightarrow K(p) \quad (5.5)$$

$K(p)$ is the cell holding the scalar value closest to the input vector parameters, p . It has been shown that in order for this property to hold, map should be less than or equal to the input dimensions of the neural network. The grid defines the spatial similarities and discontinuities for each Standard, by cluster formation. It is because of this inherent property of SOM that map organizes in such a way that any two extracted data sets close according to their spatial components are mapped onto the same cell, and with a little difference on the neighbouring cell. These parameters are extracted after analysing the received signal which may be the result of the summation of many standards, each standard itself the summation of several modulated carriers. The neighborhood function defined in equation (5.3) is the further exploitation of this property in our multi-standard transceiver design framework. The choice between SOM and LVQ is decided upon by extracted features. Then, the features identified are fed as input to trained NN, resulting in the identification of standard because of their spatial location at output grid and the property of Tonotopy.

After recognizing the standard, the receiver switches to that specific standard mode to detect the right demodulation software or download the software in a self organized manner.

Figure 37 illustrated our approach showing the architecture of our NNs, SOM, LVQ and MLP which includes:

1. The *input layer* representing the space of inputs X of dimension n .
2. The *hidden layer* models the competition space of inputs.
3. The *linear layer* of decisions.

4.3 MLP Implementation:

MLP Supervised NN is used when the desired output is known. As shown in Table 13, the huge data range (> 30 km) requires an estimation of power level, so in the case of huge data range MLP is selected to switch the CR Transceiver in IEEE 802.22 mode. The process of MLP algorithm is defined as follows:

1. Initialize the weights by small random values.
2. Propagate the input in the hidden layer and calculate the corresponding output as,

$$y_n = f(E_n) = f\left(\sum_{i=1}^m (X_i * W_{n,i})\right) \quad (5.6)$$

where m is the number of inputs coming from input layer, which are 23 in our case and X is the input vector.

3. Propagate the input in the output layer and calculate the corresponding output as,

$$y'_n = f(E'_n) = f\left(\sum_{i=1}^n y_i * W'_{n,i}\right) = a * E'_n + b \quad (5.7)$$

where n is the number of inputs coming from hidden layer, which are 10 in our case and y is the vector coming from hidden layer.

4. Calculate output layer error as,

$$\frac{\delta E_p(k)}{\delta y_n^l(k)} = -(y_n^d(k) - y_n(k)) \quad (5.8)$$

5. Calculate hidden layer error as,

$$\frac{\delta E_p}{\delta W_n'} = \frac{\delta E_p}{\delta y_n'} f'(E_n') y_n = -(y_n^d - y_n) f'(E_n') y_n \quad (5.9)$$

6. Adjust the weights by back propagating the observed error.
7. Repeat steps from 2 to 6 until the error becomes inferior to a certain threshold. We opt for sigmoid threshold function.

This procedure defined above is further simplified using the delta rule for output layer and the chain rule for partial derivatives as the error observed is calculated using gradient method. Thus, the simplified procedure is depicted in Figure 39. The implementation parameters for our designed MLP algorithm shown in Figure 39 are chosen after a careful analysis using trial and error, e.g. to avoid the undesirable situation of overfitting apart from meeting the real time constraints of our platform while parallelizing and mapping the algorithm on it.

Inputs: W, X, W', a, b
Results: Y_n', w, w'
Initialization: $a=0.25, b=0.5, j=1, i=1, K=1, L=1, n=10, m=23, p=8$

1. For $j = 1$ to n do
2. For $i = 1$ to m do
3. $E_j = \sum X_i * W_{ji}$
4. $y_j = a * E_j + b$
5. $\Delta W_{ji}(n) = \eta e_j(n) * y_j(n) * [1 - y_j(n)] * \sum_{i=1}^j [\delta_j(n) * W'_{ji}(n)]$
6. End For
7. For $k=1$ to p
8. $E'_k = \sum y_l * W'_{kl}$
9. $y'_k = a * E'_k + b$
10. $\Delta W'_{ki}(n) = \eta e'_k(n) * y'_k(n) * [1 - y'_k(n)] * \sum_{i=1}^k [\delta'_k(n) * W'_{ki}(n)]$
11. End For
12. End For
13. $R_k = y'_k ; W ; W'$

Fig. 39. Iterative MLP Algorithm for Standard and transmission power determination and weight update.

Our algorithm design consists of i input entries (1 to m), j hidden layer neurons (1 to n) and k output layer neurons (1 to p). Each layer has its own weight matrix: W_{ji} for hidden layer and W'_{ji} for output layer. The first designated index is the neuron of the layer, while the second specifies the number of entry. e_j represents output of each neuron of hidden and output layer. L represents the number of hidden layers. The sigmoid threshold function is used as it provides a bounded output and is differentiable and real-valued.

In our designed algorithm in Figure 39, from lines 2 to 6, the loop calculates the error E_j of each neuron of hidden layer as well as new weight matrix. From lines 7 to 11, the loop calculates the error E_k of each neuron of output layer as well as new weight matrix. The parameters of the above designed algorithm are:

- W (weight matrix of hidden layer formed by $n \times m$ elements).
- X (input vector to our MLP Algorithm) which are actually the parameters detected from the received signal.
- W' (weight matrix of output layer formed by $p \times n$ elements).
- Y_n (desired output of each neuron at output layer).

For hidden layer and output layer neurons, the parameters shown in Figure 39 are,

- n (designated to the current data vector being treated).
- Indexes i and j (designates respectively a neuron of previous layer and neuron of current layer).
- Index k (serves to designate a neuron in the next or following layer).

MLP is different from SOM and LVQ as it is a Supervised NN scheme and so there has to be some metric that can tell about the convergence of the algorithm to stabilize, addressing the local minima problem. We have used Mean Square Error, $E(W)$, as the cost function to measure the difference between desired output and observed output. This metric has been used successfully for classification problems using MLP in many CR applications [124]. $E(W)$ for a single presentation is defined as,

$$E(W) = \sum_{p=1}^t E_p(w) \text{ where } E_p = \frac{1}{2} \sum_{i=1}^m [y_d^i(k) - y_i(k)]^2 \quad (5.10)$$

where $y^d(k)$ is the desired vector, $y(k)$ is the obtained output value and t is the number of training examples.

4.4 Standard Recognition Experimentation Results:

This section analyses the experimental results. The training data to act as input to our NNs is based on Exponential Moving Average Algorithm, used in several modern wireless applications [120]. We have chosen hexagonal lattice of the SOM grid, *Standard*, with a size of 11×6 , which is shown in Figure 40 for standards mentioned in Table 12. We can see in Figure 40 that clusters are formed which comprise of different standards, e.g. the first two rows form a cluster of all the signals that correspond to 802.11.g standard. This means that each time a signal is encountered that translates to 802.11.g

standard, it will always be ported to this formed cluster. Each standard's cluster formation depends on the parameter values of a specific signal shown in Tables 11 and 12. The clusters are formed due to the weight vector densities across the lattice. The weight vectors having similar densities tend to occupy neighboring positions, forming a specific standard's cluster. Each cluster is identified as a

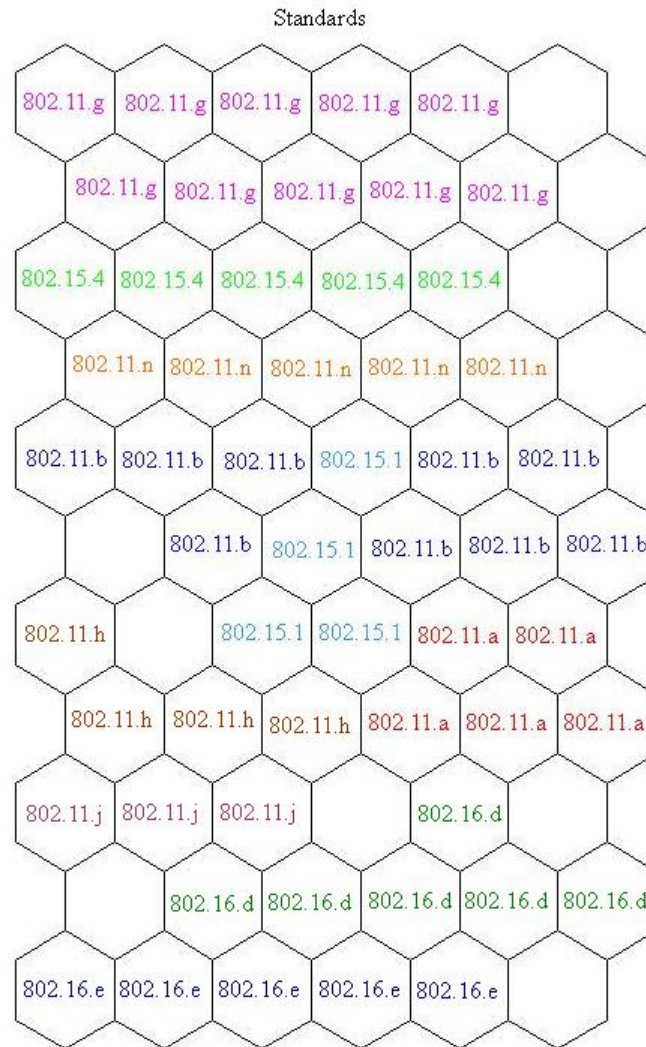


Fig. 40. Output grid showing different standards forming a cluster.

distinguishing standard, with the only exception of IEEE 802.15.1 and IEEE 802.11.b, which appear to be slightly mixed up. This can be explained because of the superimposition of the two standards at the same bandwidth of 2.4 GHz. The Final Tonotopy Error for mapping at 11 x 6 grid is 0.0133, which is an excellent result. Tonotopy error is a data dependent measure of cluster formation at the grid. It measures the input vectors for their adjacency, with respect to BMUs being neighbours. It tells the numerical deviation, at output grid, between the two BMUs of input vector. However, by solving this particular problem with respect to these two standards, we can further reduce this error value. This can be done by the addition of more parameters in Tables, e.g. the use of Wigner-Ville Transform as suggested in [129] can further discriminate in the two standards. The fact that each input signal to our

proposed SOM will map at a specific location of the output grid verifies our approach of standard recognition using the property of Tonotopy. Each location forming a cluster represents a standard and thus the cognitive radio can identify the standard based on its position at output grid. Then it can switch its operating parameters respectively to move its state to an optimized form. The visualization of the SOM standards Lattice with respect to distance between input vector and weight vector is shown in Figure 41. This U-matrix shows the distance “between” map-units as well, apart from distance “at” the map-units. As an example, the initial rows of U-matrix form a cluster that corresponds to all the signals with distances between and at map units

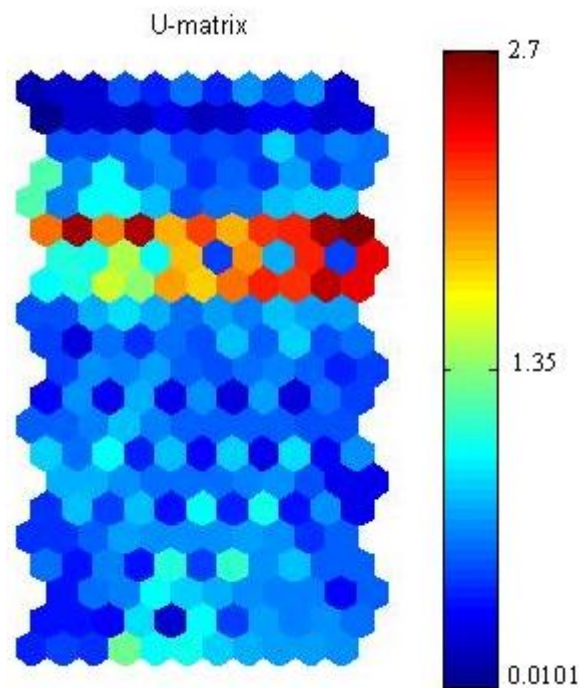


Fig. 41. U-matrix for normalized parameters for all Standards.

from 802.11.g. High values on the U-matrix mean large distance between neighboring map-units and thus indicate cluster borders, that are shown as uniform areas of low values. The colour bar at the right shows which colour means high values. Figure 42 shows a magnified view of map units in U-matrix. The three map units are IEEE 802.11.g and the same colour between the map units depicts that the distance between all three map units is constant. This constant distance between any two map-units of same standard leads to cluster formation of a particular standard.



Fig. 42. U-matrix row showing equal distance between three map units of a cluster comprising of 802.11.g Standard.

The same cluster formation is observed for the standards mentioned in Table 11. We chose to have the same dimensions of output grid (11 x 6) and so each Standard is recognized with the help of its location at the output grid. The only difference is the absence of the neighborhood function, as we have used LVQ and not SOM. Thus, for each new signal received, the map can be used to classify it.

In the NN decision phase explained earlier in Figure 36, if the data range is greater than 30 km, then the decision is in the favour of MLP. The estimated amount of transmission power needs to be matched with the target transmission power and hence the optimization of cost function explained in equation (5.10) needs to be done. We have used the Error Back Propagation algorithm for training purpose of MLP and our algorithm tends to be stable after 700 epochs. An epoch is a single

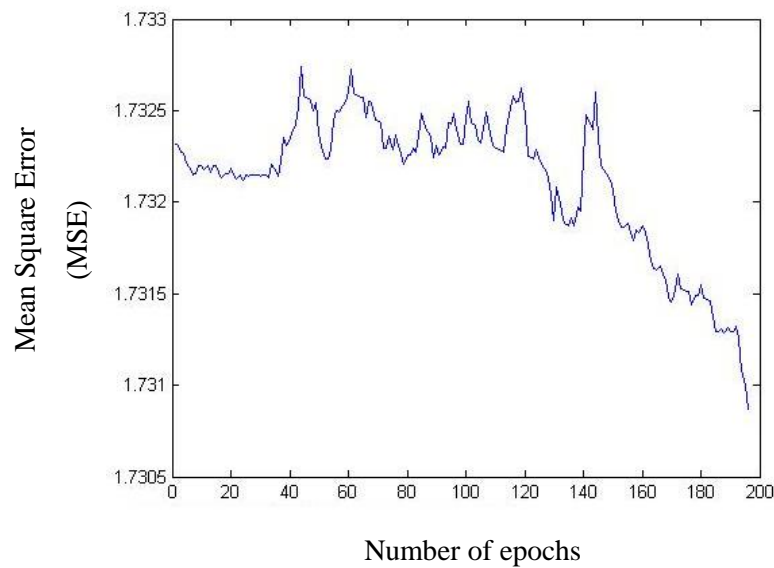


Fig. 43. MSE after 200 training epochs

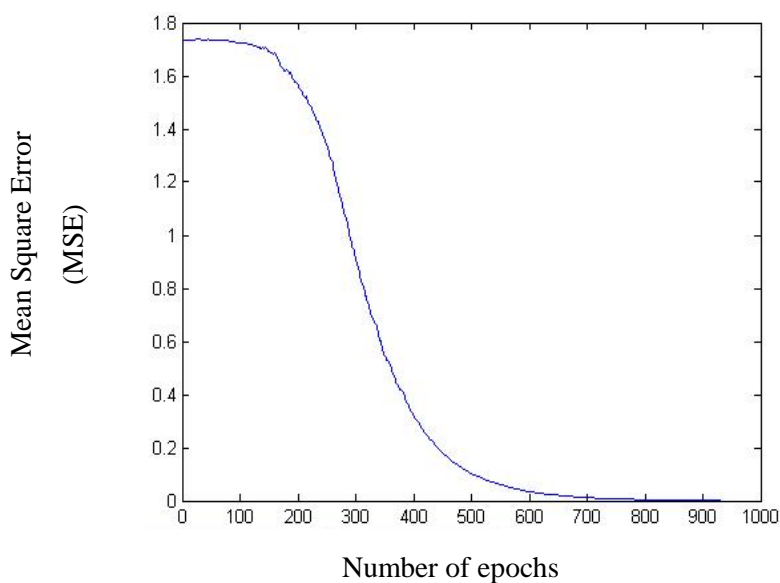


Fig. 44. MSE after 700 training epochs

presentation through the process of providing the network with an input and updating the network's weights. Like SOM and LVQ, the training data is based on Exponential Moving Average Algorithm. MSE measured for standard recognition and transmission power estimation in the case of 802.22 at the MLP network outputs after 200 and 800 training epochs is shown in Figures 43 and 44, respectively. Thus, in the case of huge data range, the standard is identified as 802.22 and its expected transmission power is matched with the observed value of transmission power after 700 epochs. The MSE observed is 0.02 which is again a very encouraging result. Thus in this section, we have proved that our designed NN algorithms: LVQ, SOM and MLP are the most efficient standard recognition solutions for multi-standard CR Transceiver, which outperforms the counterpart NN Techniques [124,120, 122, 127] used for cognition incorporation in radios. A comparison is done in Table 14 that shows the error values for the best cases by using different NN techniques.

Researcher	Neural Network Technique	Cognition purpose	Error Value
Orcay[124]	Multi Layer Perceptron	Distinguish between Signal patterns	0.0625
Tsagkaris[120]	Focused Time Delay	Data rate prediction to choose best RAT	0.0637
Baldo[122]	Multi-Layer Feed Forward	Throughput prediction of 802.11 cell to dynamically select channel	0.8
Roland[127]	Radial Basis Function	Distinguish between two standards	0.021
<i>Our approach</i>	<i>SOM and LVQ</i>	<i>Recognizing specific standard amongst a list of pre-defined Standards</i>	<i>0.0133</i>
	<i>MLP</i>		<i>0.02</i>

Tab. 14. Comparison with other NN based CR proposals

Table 14 clearly shows that our designed NN algorithms give much better results than other NN techniques proposed for cognition incorporation in SDRs.

Our proposed standard recognition approach is also different as it uses an adaptable solution that exploits three Neural Networks to recognize different standards, thereby allowing CR to switch in the specific standard mode, optimizing the radio configuration to give the best Quality of Service (QoS). We have also seen in this Section that CR Transceivers require both, high performance and high adaptability. However technological advancements in the coming years will push communications hardware into obsolescence at an ever quicker pace which poses an acquisition challenge for CR hardware device. This makes CR Transceiver algorithms implementation on embedded systems a hot issue for the rapidly developing wireless communication networks. The adaptability to varying situations and unknown systems illustrate the value of CR Transceiver algorithms on general purpose hardware against the better performing but less flexible dedicated hardware. Optimization for QoS means little if the hardware is incapable of supporting the resulting waveforms. So, there has been some efforts in this direction. M.E.Sahin [130] focuses on sensing and adaptation aspects and

performs simulations and analysis regarding the practical implementation of CR communications. These are related to the transmission of spectrum sensing via USB, enabling CR to detect a licensed system. Kim [131] presents a two step Genetic Algorithms based Software Testbed to optimize the parameters such as carrier frequency, bandwidth, transmit power and modulation type. N. Muhammad describes a flexible yet efficient hardware design for Front End processing (FEP) that includes FFT, product/division and Dot Product of SDR [132]. M.A.Cavuslu [133] used MATLAB simulations to show parallel hardware implementation of Neural Network FPGAs. Hardware/Software Co-design for embedded implementation of Neural Networks on reconfigurable devices is discussed in [134] using a software called NNetWARE-Builder. A.D. Rast [135] introduces SpiNNaker, a dedicated neural chip multiprocessor that combines an array of general purpose processors with a configurable asynchronous interconnect and memory fabric to achieve on and off chip parallelism, universal network architecture support and programmable temporal dynamics.

After identifying the efficient NN based cognitive algorithms, Section 5 describes our parallel implementation strategy of Cognitive Neural Networks based algorithms on the platform described in detail in Section 4 of Chapter 4.

5. Parallelization and Implementation Strategy:

Since our target platform contains 16 PEs, in order to map the cognition incorporated algorithms, explained in Section 4, on our target platform, explained in Chapter 4, we designed a parallelization strategy and then implemented our algorithms using our designed strategy on our target chip. As explained earlier, the three designed algorithms are inherently parallel because of their biological origin and thus highly suitable for mapping on our target platform, changing our SDR waveform into Cognitive waveform. This Section explains our parallelization strategy in terms of efficient load balancing and the implementation of the NN based cognition algorithms exploiting our designed strategy.

5.1 SOM and LVQ Parallel Implementation:

Since LVQ and SOM have a very similar algorithm design, we describe our designed parallelization strategy for LVQ. The SOM implementation follows the exact strategy with additional computation of the neighborhood function and corresponding weight updates. Our parallelization strategy for LVQ starts with the Master PE executing various control tasks for the slave PEs. Each PE has a unique identity called as *id*.

If the total number of PEs is p , we divide the n neurons computation amongst p PEs. We have ported the LVQ (and SOM) four times changing the number of PEs each time to judge the effects of NoC

characterized by bandwidth and latency. Each time, we have configured our platform architecture to include 1, 4, 8 and 16 PEs respectively. The number of neurons remains 23 each time. We have chosen n to be 23 after a careful analysis using trial and error method to avoid the undesirable situations of overfitting and network overload. In the first case, when there is a single PE, all the neurons are handled by the same PE in a sequential loop, one after the other. However, for the other three cases, we have designed special primitives that we explain hereafter: After control and memory initialization, the Master PE gives a start command to the slaves using *Syn_start_work*(). Slave processors, on receiving the Master PE command, who were in waiting state using *Syn_wait_for_start*(), start their computation for their respective Euclidean distances, then the positive and negative weights respectively. Each slave, after finishing, puts the results in DDR accessible to it via Data NoC, and writes a flag in synchronization memory using *barrier*(). Using Synchronization NoC, Master PE checks if all slaves have finished their computation by checking the flag value using *Syn_work_finished*(). Once the flag indicates the end of all slave computations, Master PE determines the winning neuron, updates its weight and assigns it the label.

Several neurons are handled by the same processor in a sequential loop as the number of processors are less than the number of neurons ($p < n$) in all four cases. Following the single PE case, where all the

```

1. For (j = id; j < k; j += N_Processor)
   Parallelization Start Primitive
2.   For i = 1 to n do
3.     
$$d_j = \sum_{i=1}^n |X_i - W_{ij}|^2$$

4.     
$$W_j^+(t+1) = W_{ij}(t) + h_{ci}(t)[X_i(t) - W_{ij}(t)]$$

5.     
$$W_j^-(t+1) = W_{ij}(t) - h_{ci}(t)[X_i(t) - W_{ij}(t)]$$

14.   End For
15.   If ( $d_j < d_{min}$ )      Then  $E_G = E_{pj}$ 
16.                        $d_{min} = d_j$ 
17.                       If ( $E_{pj} = E_f$ ) Then  $W_G = W_j^+$ 
18.                       Else  $W_G = W_j^-$ 
19.                       End if
20.   End if
   Parallelization Finish Primitive
21. End for

```

Fig. 45. LVQ and SOM parallelized Algorithm

computation is performed sequentially, we have divided load evenly amongst all the PEs, i.e. each PE simulates n/p neurons if n is perfectly divisible by p , otherwise the remainder is distributed again amongst some of the PEs. In the case of SOM, the only difference is the weight updating and assignment of label to the neighboring neurons as well, defined in the vicinity of the winning neuron by the help of Gaussian neighborhood function defined in equation (5.3). We illustrate the changes that we made in our algorithm in Figure 45. $N_Processor$ is the number of PEs configured to compute.

5.2 MLP Parallel implementation:

We have already observed in the MLP algorithm in Figure 39 that it does not offer that high degree of internal parallelism. Furthermore this algorithm performs quite an extensive communication between the neurons after each iteration. So, the parallelism of MLP is a bit tricky as the well known problem of sticking at local minima is encountered if careful load balancing is not done. Like LVQ and SOM, we have ported the MLP four times on our target platform, changing the number of PEs to 1, 4, 8 and 16, respectively. For the number of PEs 4, 8 and 16, we have parallelized the MLP in two steps: The first step deals with hidden layer neurons and the second with output layer neurons. In the first phase, the Master PE gives a start command to the slaves using *Syn_start_work* (). The equal load balancing between all the processors is done by changing the loop primitives according to the number of PEs configured to compute, where *id* is identity of the specific processor that will carry out the task, *k* is

```

1. For j = 1 to n do
2.   For (i=id; i<k; i+=N_Processor)
3.     Parallelization Start Primitive 1
4.     
$$E_j = \sum X_i * W_{ji}$$

5.     
$$y_j = a * E_j + b$$

6.     
$$\Delta W_{ji}(n) = \eta e_j(n) * y_j(n) * [1 - y_j(n)] * \sum_{i=1}^j [\delta_j(n) * W'_{ji}(n)]$$

7.     Parallelization Finish Primitive 1
8.   End For
9.   For (h=id; h<k; h+=N_Processor)
10.    Parallelization Start Primitive 2
11.    
$$E'_k = \sum y_h * W'_{kh}$$

12.    
$$y'_k = a * E_k + b$$

13.    
$$\Delta W'_{ki}(n) = \eta e_k(n) * y'_k(n) * [1 - y'_k(n)] * \sum_{i=1}^k [\delta_k(n) * W_{ki}(n)]$$

14.    Parallelization Finish Primitive 2
15.  End For
16. End For

```

Fig. 46. MLP Parallelized Algorithm

the number of neurons and *N_Processor* is the number of PEs used for the computation. The slave processors start their computation for error E_j as well as new weight matrix. Once this is done, all PEs put the result in DDR and a flag is written in the synchronization memory. The end of the first phase is determined by the flag indication of the end of all slave computations. Before the start of the second phase, we have to manually reset the flag variable in synchronization memory accessible via *Synchronization.h*. This is meant to tell the Master PE about the start of the second phase.

In the second phase, the error E_k' of the output layer neurons and weight matrix is calculated. After

receiving the Master PE command, all slave PEs start their respective computations as in the first step and then put the results in external DDR memory using the same primitives.

The two phases are closed in a loop that tends to repeat all the above defined procedures until the desired threshold of MSE, explained in equation (5.10) and Figures 43 and 44 is not obtained. Like SOM and LVQ, we have ported MLP four times to judge the effects of NoC characterized by bandwidth and latency. The illustration of changes made in our algorithm is depicted in Figure 46, with both the phases.

The parallelization strategy for our designed algorithms: LVQ, SOM and MLP meant to recognize the Standard in our CR multi-standard Transceiver, together with the parallelization primitives designed for the MPSoC general purpose platform is illustrated in detail in Figure 47.

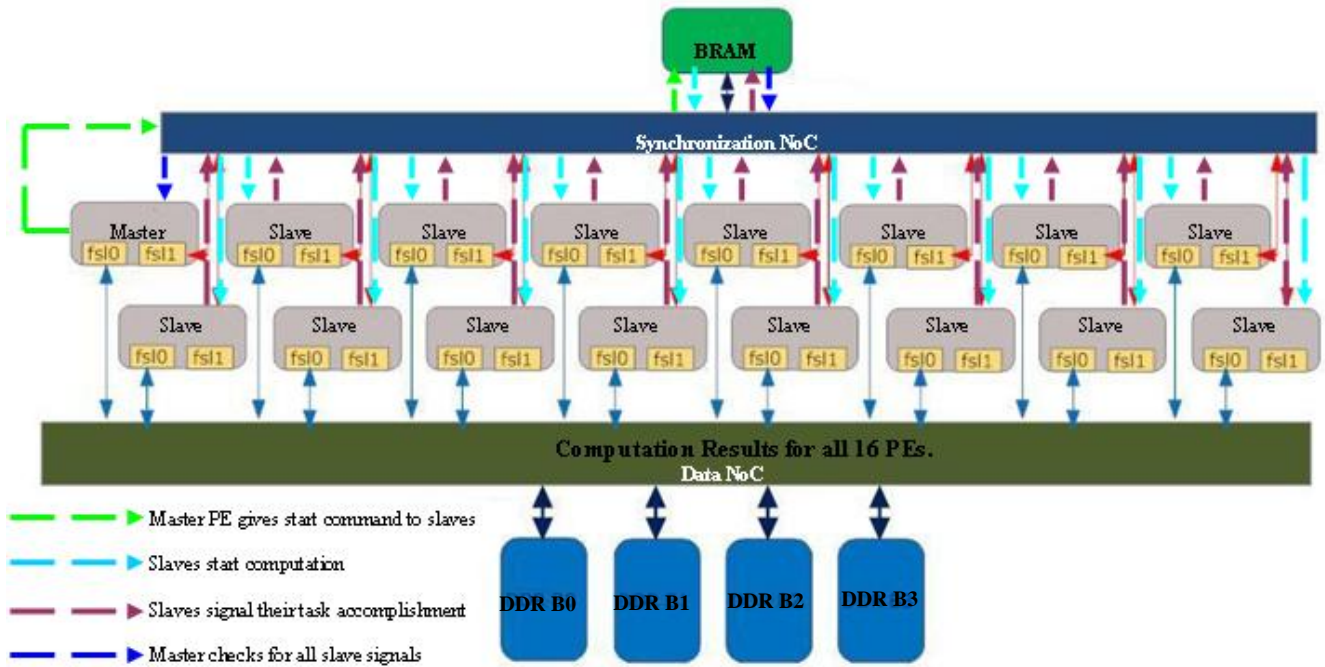


Fig. 47. Parallelization illustration with the primitives.

6. Speed up Results:

This section evaluates the performance, in terms of speed up, of our Neural Networks based CR Standard recognizing algorithms on our designed platform. Speed up is the ratio between the number of clock cycles needed for computation on a single PE and number of clock cycles needed for computation on multiple PEs. The number of clock cycles and speed up for each of the three algorithms: LVQ, SOM and MLP exploited in the framework of standard recognition in a CR context to engineer an universal transceiver is shown in Table 15.

Cognition Algorithm	Single PE clock cycles	Speed up with 4 PEs	Speed up with 8 PEs	Speed up with 16 PEs
LVQ	1,486,899	3.72	7.68	14.39
SOM	1,667,911	3.45	7.18	14.13
MLP	2,087,142	3.12	7.01	13.79

Tab. 15. Speed-up of ANN Cognitive Algorithms using Multiprocessor platform.

As explained in Section 5 and shown in Table 15, each of these three algorithms is ported four times changing the number of parallel PEs, each time. Table 15 shows that LVQ, SOM and MLP have respected speed ups of 14.39, 14.13 and 13.79 if we exploit 100 % resources of the designed chip, i.e. when all 16 PEs are configured to perform their respective computations. However, we observe that the speed-up diminishes slightly with the addition of more processors. This can be attributed to the communication load characterized by data throughput and latency to establish a connection between two processor nodes for data exchange. Since the received signal may be summation of many standards and each PE in our chip has the capability to execute its assigned tasks independently of the other fifteen PEs, we can overcome the limitations that can occur when adding more processors by dividing the 16 PEs in further independent groups. Each group can work on a different standard and thus a received signal with more than one standard can be processed. This way the performance efficiency of our proposed CR Transceiver can be further exploited and a complex signal composed of many modulated carriers can be processed, e.g. we can divide 16 PEs in three independent groups of 8, 4 and 4 PEs, followed by the porting of MLP on the first group of 8 PEs and porting SOM and LVQ respectively on the remaining two groups of 4 PEs, each. This means that we can even increase the number of PEs from 16 and can port other SDR resources as well, to constitute a complete Cognitive SDR waveform. Thus, the area performance trade-offs for efficient design of Cognitive Radio waveform is addressed in the most efficient way possible.

There is little difference between the obtained speed up of LVQ and SOM, which can be attributed to the fact that LVQ needs to update only the distance of winning neuron in contrast to SOM that requires additional computation to update the distance of all neighboring neurons apart from calculating the neighborhood function. In the MLP algorithm, the speed up shown is the aggregate speed up obtained after the execution of both the phases on the designed chip. The speed up obtained from all three algorithms is visualized in Figure 48. We have seen in Table 15 that applying our original solution of Neural Networks based Cognition Incorporation Algorithms on our designed general purpose MPSoC provides significant speed up, thereby allowing us to avoid expensive hardware accelerators. Furthermore, our proposals to divide the number of available PEs into independent sub-groups to treat multi-standard signals can be implemented according to end user requirements and the specific application needs.

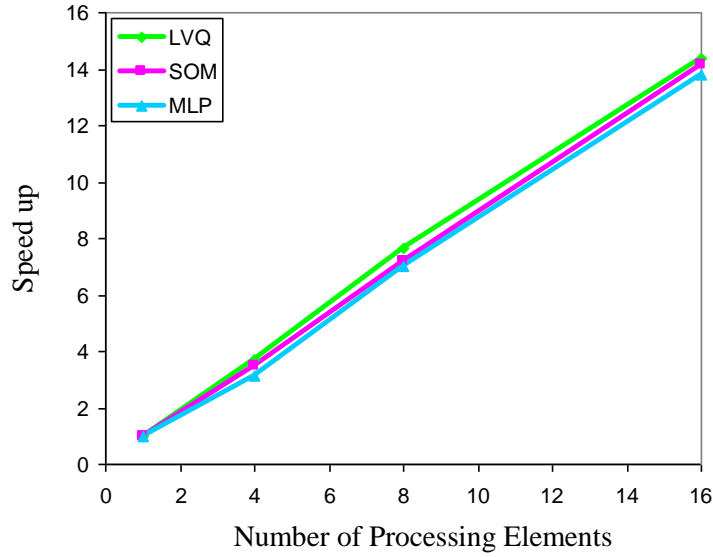


Fig. 48. Speed-up obtained for all three Neural Networks based cognition algorithms.

7. Conclusions:

The future of wireless communications will be characterized by highly varying environments with multiple diverse features. In such an unfamiliar scenario, CRs need some efficient information processing algorithms. Artificial Neural Networks are widely different from conventional information processing as they have the ability to learn from given examples thus being able to perform in cognitive tasks by exhibiting the basic instinct of CR, explained in Cognition Cycle. In this respect, we proposed a multi-standard CR Transceiver based on three NN Techniques: LVQ, SOM and MLP, which are widely different from conventional information processing. Using LVQ, SOM and MLP, our proposed CR Transceiver recognized the standard based on parameters received as input to NN, moving the radio in the recognized standard mode to make adequate decisions. This leads to optimize radio resource use (spectrum, battery, carrier frequency). The experimental results clearly show that our proposed solution outperforms other proposals, as shown in Table 14.

At the same time, we also exploit the inherent parallelism in mentioned Neural Networks Techniques to answer the high adaptability requirements of CR embedded implementation by mapping the proposed CR Transceiver Algorithms on the designed MPSoC. Table 15 reminds us the speed up limitations that can occur while adding more PEs. We address such limitations by partitioning the PEs into different groups, each group dedicated to identify a different standard of received signal, which may be the result of summation of many standards, and thus further optimizing the resource utilisation of the designed MPSoC based CR Transceiver.

Thus, we proposed and implemented an efficient solution to cognition incorporation in our SDR

designed waveform. After identification of specific standards to optimize the radio resource usage, our future work will address the prediction of future occupancy trends of the radio resource (Spectrum), using Neural Network Techniques, that we address in Chapter 6. In today's Spectrum constraint environment, it is essential to undergo a radical rethinking, that could enable an optimized spectrum usage. Chapter 6 explains our strategy to address this optimized Spectrum usage.

Chapter 6:

Cognitive Radio Spectrum Evolution Prediction using Artificial Neural Networks based Time Series Modeling.

1. Introduction:

CR as defined in Chapter 3 is an efficient answer to spectrum scarcity as it can sense the spectrum steadily based on previous information about the spectrum evolution in time, thus predicting the future occupancy status. Framed within this statement, this Chapter proposes a new methodology for spectrum prediction by modelling licensed signal Radio Frequency (RF) features as a multivariate chaotic time series, which is then given as input to Artificial Neural Network, that predicts the evolution of RF time series to decide if the unlicensed user can exploit the spectrum band. We exploit the inherent cyclostationarity in primary signals for Non-linear Autoregressive Exogenous (NARX) Time Series Modelling of RF features, which is an extremely challenging task due to interdependence of different RF features.

Dynamic and opportunistic spectrum access is not a new idea and is probably as old as radio communications itself. The idea of shared spectrum or radio resources dates back to 1920's, when the band of 2 KHz was used as an emergency communication meant for ships communications. In 1970's Federal Communications Commission (FCC) authorized the shared communication at the civil band of 27 MHz [182]. However, it was required that the maximum transmit power limits for the civil band should be respected. Then the beginning of wireless technology and data communication saw smarter ways of spectrum management. Today, Spectrum resource is seen as, an inexhaustible but limited in its usage, natural radio resource. In order to make sure that this resource is being used in an economical and fair way, national and international controlling entities such as Pakistan Telecommunication Authority (PTA), International Telecommunication Union (ITU) and Federal Communications Commission (FCC) manage the commercial and public radio spectrum in their respective geographical areas. Their task is to ensure the interference free operations by making sure

that the spectrum users are abiding by their radio regulations. Actual wireless communication systems exploit the static radio resource

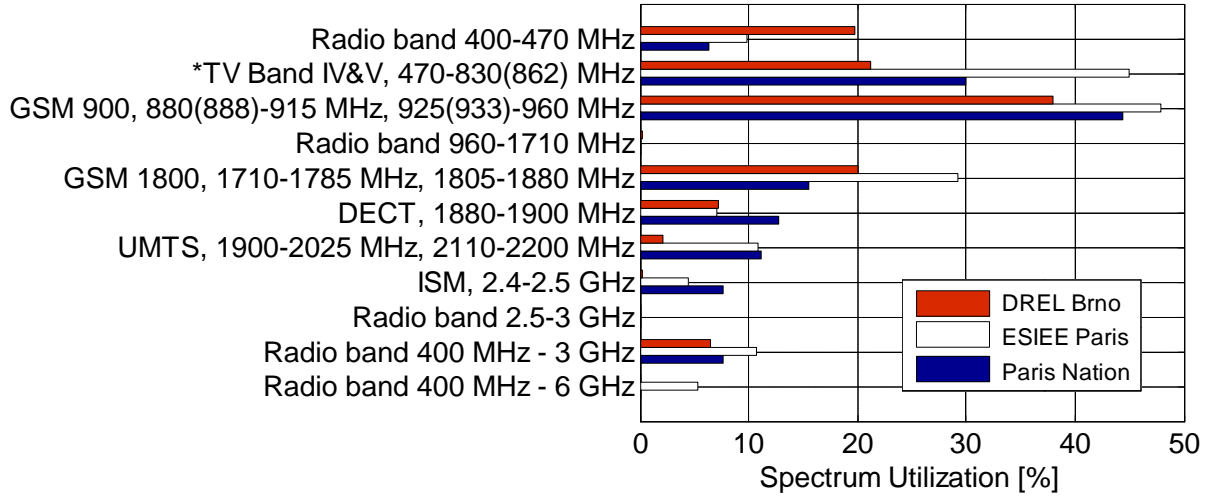


Fig. 49. Recent Spectrum Utilization in Percentage in Europe (taken from [136]).

allocation, i.e. a certain portion of the radio spectrum resource covering a large geographical area is licensed to a specific party for a long term. This is a beneficial approach in the sense that it gives an extreme protection from harmful interferences in any specific allocated radio band. However, there are many recent measurement studies [136,137] that reveal that this fixed RF allocation leads into significant underutilization of this precious resource as there is very limited exploitation in different geographical regions in different time intervals. This underutilization is depicted in Figure 49. CR addresses this paradox of spectrum scarcity by detecting and occupying spectrum holes intermittently. However, regulators need to insure that secondary users (non-licensed holders) will not interfere with the primary user (license holder) of the considered spectrum band. In this Chapter, we target such interference avoidance by predicting the future spectrum occupancy pattern of primary user on the basis of previous occupancy observations.

Mathematically, if H_0 and H_1 correspond to primary user signal absence and presence respectively, then the goal is to distinguish between following two dynamic hypothesis, on the basis of previous observations, at a given instance t :

$$\left. \begin{aligned} H_0 : x(t) &= n(t) & 0 < t \leq T \\ H_1 : x(t) &= s(t) + n(t) & 0 < t \leq T \end{aligned} \right\} \quad (6.1)$$

where T denotes the observation time, $n(t)$ is the Additive White Gaussian Noise (AWGN) with zero mean. $x(t)$ is the received signal, $s(t)$ is the deterministic complex transmitted signal whose mean and autocorrelation exhibit periodicity.

In order to decide between the two hypotheses, we model the primary user spectrum occupancy evolution as a NARX RF time series [138]. However, the cyclostationary features of a deterministic

signal (different variables of a NARX RF time series) affect each other which makes it difficult to attain global minima [139]. We overcome this problem by Artificial Neural Networks which have the potential to predict the spectrum occupancy dynamic evolution by accepting primary user RF features modelled as a chaotic NARX time series as input and giving the future expected spectrum evolution as output. We propose to predict the spectrum occupancy by modelling RF time series using Elman Recurrent Neural Network (ERNN) [140]. The only difference between Recurrent NN and Feed-forward NN is the presence or absence of feedback connections that extend from output of neurons to previous layer input neurons. Multi-layer Perceptron [105] is the most common example of Feed-forward NN. We opt for Recurrent NN as they preserve a sense of history which means that previous state RF features form a part of current state inputs. The motivation to use ERNN for our RF time series modelling to predict the spectrum evolution in time comes from their inherent sigmoid hidden layer neurons, explained in our problem context in detail in Section 5 of this Chapter.

There has not been any effort in spectrum prediction using Recurrent NN modelling of RF NARX time series. The work presented in this Chapter is different as we predict the spectrum evolution by exploiting the cyclostationary signal features to construct a RF NARX time series that contain more information than the univariate time series [141]. This is in contrast to most of the modelling methodologies which focus on the univariate time series prediction [142]. The opportunistic spectrum access by cognitive users is visualized in Figure 50.

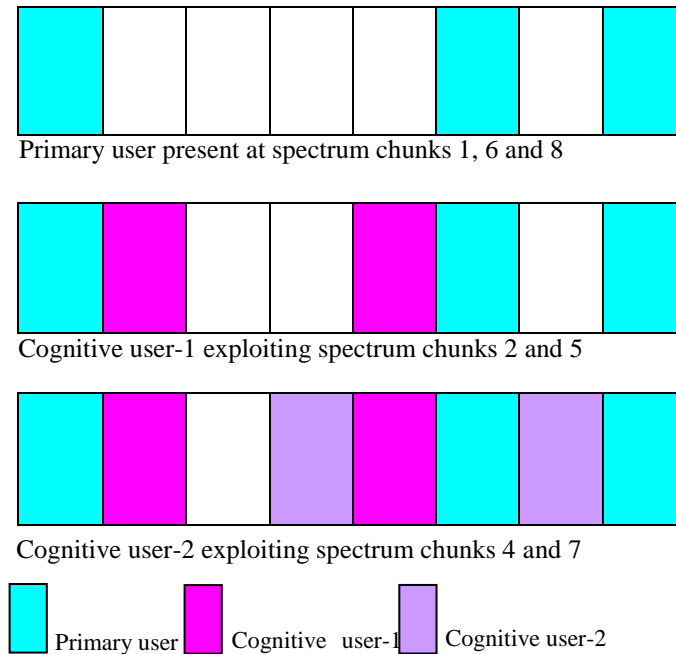


Fig. 50. Opportunistic Spectrum access by cognitive users.

2. Spectrum Opportunity Sensing Methods:

The spectrum opportunity concept is the notion of secondary users recognizing unused frequency bands by primary or licensed users and exploiting those bands as an available opportunity. The exploitation refers the transmission over the recognized unused frequency bands until the licensed user becomes active. The unused frequency band or channel is referred to as a spectrum opportunity [143,144,145]. This concept is formally defined in literature as: *A channel is an opportunity to a secondary transmitter A and secondary receiver B if they can communicate successfully over this channel while limiting the interference to primary users below a prescribed level determined by the regulatory policy.* The essential requirement is that the secondary users should vacate the *opportunity* at the earliest, after the primary user appears. Also, to ensure low interference at the primary receiver side, the detection sensitivity of the cognitive user has to be reasonably high.

In the following Sub-sections, an overview of spectrum sensing methods proposed in the literature is presented. Moreover, we explain our motivation to exploit cyclostationary feature detection technique. The spectrum sensing algorithms proposed in the literature may be broadly divided to following categories defined in this section.

2.1 Matched Filter:

Matched Filter [146] is the optimal way for detecting a known signal in the presence of additive Gaussian noise. The reason for this feature is its characteristics that maximize the SNR. It is matched to some periodically repeated but unknown component of a signal, hence called matched or coherent detector. It is the linear filter, whose output is given by,

$$y = s^H \sum_n^{-1} x \quad (6.2)$$

where x is the observation vector, s is the known deterministic signal to be detected and \sum_n is the noise covariance matrix.

If, the noise is Gaussian, then the output ‘ y ’ in equation (6.2) is Gaussian as well. The reason is that it is the linear transformation of a Gaussian random vector. The output’s (y) mean is zero under H_0 . The implementation of this technique is constraint by the waveform’s prior information, such as, pilot signals and preambles. This fact is also evident from equation (6.2), as we can see that it requires the knowledge of the transmitted signal s and the noise covariance matrix \sum_n . In addition, the synchronization errors may degrade the performance severely. The only advantage is its easy design and general applicability.

2.2 Energy Detection:

Energy detector [147] based approaches are also known as radiometry or periodogram. In this method, the received energy of the signal is compared with a threshold. The energy detector is explained by:

$$y_M = \frac{2}{N_0} \sum_{n=1}^M |x(n)|^2 \quad (6.3)$$

where $x(n)$ is the received complex valued discrete time signal, N_0 is the noise power and M is the number of observations. Factor 2 explains the fact that complex noise power is equally bifurcated between the real and imaginary parts, under circularity assumption. Figure 51 depicts the block diagram of the energy detector.

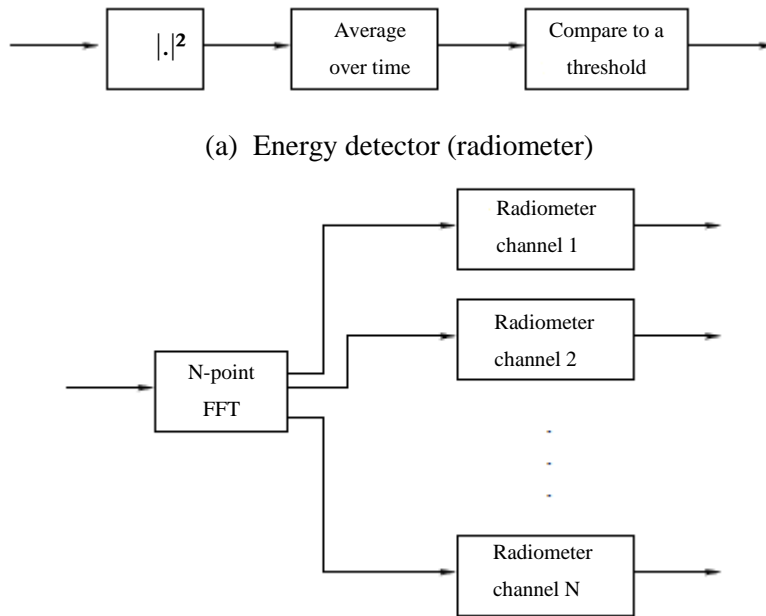


Fig. 51. Block diagram of (a) the basic energy detector and (b) an energy detector in frequency domain.

In energy detection based algorithms, the threshold used is dependent on the noise variance. Therefore, threshold determination is very important. A large threshold may lead to overlook an activity and small threshold may result in false alarms due to background noise. This method is called as optimum detection method because of two reasons. (1) It has low computational and implementation

complexities. (2) The sensing time is faster. The generic design of this method does not require any information about primary users. The optimum detection properties make the optimum detection of random uncorrelated Gaussian signal [152]. However, there are some disadvantages associated with this method. Some prior reliable estimate about noise power is required, otherwise significant performance loss is observed. Also, this method is unable to distinguish between interference, primary user and secondary user. The third inconvenience is addressed above that talks of the importance of threshold determination. Another known fact is the performance degradation under Rayleigh fading. The coexistence of UWB and WiMAX has been addressed by energy detection of WiMAX systems in [149]. The authors in [150] identify the idle slots in GSM signal, using energy level measurement, for further exploitation.

2.3 Other approaches:

There have been some more methods proposed for spectrum sensing such as waveform based sensing [151], classic likelihood ratio test [152], Eigenvalue based sensing and Covariance based sensing [106]. The likelihood ratio test method requires both source signal and noise power information. Wavelet based sensing methods require only noise power information (semi-blind detection), whereas

Spectrum Sensing Methods	Limitations
Energy Detection	Poor performance under low SNR
	Inefficient for spread spectrum signal detection
	Sensing performance is degraded due to noise uncertainty
Matched Filter Detection	Huge power consumption
	Implementation complexity
	Requires primary user perfect prior knowledge.
Waveform based sensing	Susceptible to synchronization errors
	Signal power estimation is difficult
Classic Likelihood ratio test	Requires source signal and noise power detailed information
Newly Emerging Methods (a) Eigenvalue based sensing (b)Covariance based sensing	Does not consider periodic features of a signal

Tab. 16. Spectrum Sensing methods and their limitations.

newly emerging methods such as eigenvalue based sensing and covariance based sensing require no information on source signal or noise power (totally blind detection). All the methods discussed until now are summarized with their limitations in Table 16.

There is another method called Cyclostationary Detection [148, 153] which is based on stationary process statistics over some period τ that we have exploited. This method overcomes the limitations mentioned in Table 16 due to robustness in noise power and propagation channel. We explain in detail this method and our motivation to exploit it in detail in the following Sub-section.

2.4 Cyclostationary Detection:

The modulated signals have some special features which cause them to exhibit periodicity related to symbol periods, cyclic prefixes, hopping rates and other such features. These non-random components make the practical communication signals look like a stationary process as the statistical characteristics of the modulated signals: mean and autocorrelation exhibit cyclic or periodic features. Therefore, they are characterized as cyclostationary feature possessing signals, which interprets that their above mentioned statistical parameters vary periodically in time. For a cyclostationary signal, the Spectral-correlation density (SCD) function is known to take nonzero values and nonzero cyclic frequencies. At the same time, noise does not possess these statistical features, which means that its SCD is always zero at all nonzero cyclic frequencies. Thus, the signal can be easily distinguished from noise by SCD analysis. In addition to that, we can distinguish the signal type as different signals have different non-zero cyclic frequencies. Cyclostationary detection can also differentiate between different modulation schemes as each scheme exhibit different SCD. In the following, we list cyclic frequencies for some signals that can be of potential interest. We have taken these cyclic frequencies from [154]

1. Analog TV signal: It has cyclic frequencies at multiples of the TV-signal horizontal line-scan rate. (15.6 KHz in Europe).

2. AM signal: $x(t) = a(t)\cos(2\pi f_c t + \phi_0)$. It has cyclic frequencies at $\pm 2f_c$.

3. PM and FM signal: $x(t) = \cos(2\pi f_c t + \phi(t))$. It has cyclic frequencies at $\pm 2f_c$. The characteristics of the SCD function at cyclic frequency $\pm 2f_c$ depend on $\phi(t)$.

4. Digital Modulated Signals:

i. Amplitude Shift Keying:

$$x(t) = \left[\sum_{n=-\infty}^{\infty} a_n p(t - n\Delta_s - t_0) \right] \cos(2\pi f_c t + \phi_0).$$

It has cyclic frequencies at k/Δ_s , $k \neq 0$ and $\pm 2f_c + k/\Delta_s$, $k = 0, \pm 1, \pm 2, \dots$

ii. Phase Shift Keying:

$$x(t) = \cos \left[2\pi f_c t + \sum_{n=-\infty}^{\infty} a_n p(t - n\Delta_s - t_0) \right].$$

ii(a). BPSK has cyclic frequencies at k/Δ_s , $k \neq 0$ and $\pm 2f_c + k/\Delta_s$, $k=0, \pm 1, \pm 2, \dots$

ii(b). QPSK has cyclic frequencies at k/Δ_s , $k \neq 0$.

When source signal $x(t)$ passes through a wireless channel, the received signal is impaired by the unknown propagation channel. In general, the received signal can be written as,

$$y(t) = x(t) \otimes h(t) \quad (6.4)$$

where \otimes denotes the convolution, and $h(t)$ denotes the channel response. It can be shown that the SCD function of $y(t)$ is

$$S_y(f) = H \left(f + \frac{\alpha}{2} \right) H^* \left(f - \frac{\alpha}{2} \right) S_x(f) \quad (6.5)$$

Where $*$ denotes the conjugate, α denotes the cyclic frequency for $x(t)$, $H(f)$ is the Fourier transform of the channel $h(t)$, and $S_x(f)$ is the SCD function of $x(t)$. Thus, at certain cyclic frequencies, the unknown channel could have major impacts on the SCD strength.

If we compare the cyclostationary feature detection with other methods described above, we will see that this method provides the best performance in sensing situations where characteristics of signal, channel and noise are known before hand. Another important advantage is the robustness to uncertainty in noise power and propagation channel as it offers the best performance under low SNR. Despite the advantages, there are certain inconveniences associated with this method. It is highly susceptible to ADC sampling clock offsets, so the sampling time could affect the cyclic frequencies. If we try to overcome this problem by using oversampling, it will introduce another drawback of high sampling rate requirement. Besides, the computation of SCD function is a computationally complex issue.

Cyclostationary feature detection inherently suits our econometric technique of primary signal RF features modelling using NARX time series. This is explained by the fact that noise does not have any cyclostationarity in contrast to primary user signal that has special detectable statistical or cyclostationary features. Consequently, in order to predict one feature at a future time instance, to determine the presence of primary user, we need to consider other relevant previously observed RF features. This combined modelling notion is known as Non-linear Autoregressive Exogenous Model in time series literature [155].

3. Multivariate Time Series Prediction and Neural Networks:

Time Series Prediction has proved to be one of the most important problems in many computing applications because of its wide ranging applications. In our context, we apply time series prediction to optimize the spectrum usage. In any case, the prediction results are based on knowledge of some aspects of previous observations of the system. The prediction model to be implemented is determined by the complexity of time series or by the problem to be modeled as time series. The traditional prediction methods: such as Auto-Regressive Moving Average (ARMA) and Auto-Regressive Integrated Moving Average (ARIMA) [156] are the statistical approaches to model time series. Markov Models have also been used to perform time series prediction [157]. The most powerful method known to predict time series is Artificial Neural Networks. The Artificial Neural Networks can be classified into two classes with respect to their topologies: Feed-forward and Recurrent. The dynamic systems such as Spectrum Evolution in time, are composed of many states and the spectrum evolution evolves according to many non-linear relationships between RF features, represented by equations. Such dynamic systems are well-represented by The Recurrent Neural Networks. The equations help to keep a record of the previous history and predict the future expected trend because of this knowledge. It is because of this reason that Recurrent Neural Networks are more suitable in prediction requiring problems. In recent years, many recurrent neural networks such as, Simple Recurrent Network (SRN) and Simple Recurrent Network with Shortcut Connections (SRNSC) have been investigated to address the single step ahead prediction problem. The recurrent inputs of the SRN architecture are connected to hidden neurons, which is the kind of network that we have exploited for solving our problem (Elman Recurrent Neural Network). The recurrent links (connection between the hidden layer) help to keep a history of previous occurred events, therefore, this model suits our application as we want to predict the Spectrum Evolution in future based on previous observations. The SRNSC architecture is also a class of SRN model. However, the connections are between input and output neurons. Many Recurrent Neural Networks architectures such as the Nonlinear Autoregressive with Exogenous Input (NARX) model [158, 159] and a dynamic recurrent network are presented by Parlos [160] for the multi-step ahead prediction. He proposed a novel recurrent multilayer perceptron architecture based on a modified learning algorithm. The connections in any dynamic recurrent neural network consist of mainly three kinds of links: feed forward links, recurrent links and cross-talk links. Recurrent links are the connection between the neurons of hidden layer and the input layer, whereas cross-talk links are the connection between neurons of the output and hidden layer. In order to improve prediction performance, many efforts have been done to propose novel architectures for neural network based prediction. As an example, Owens [161] compared the Feed Forward Nonlinear Autoregressive Model, the fully recurrent architecture and many other NN architectures for determining the best prediction results. In [162], the authors present a novel algorithm to train the neural network that identifies chaotic dynamics from a single measured time series. The

work presented in [163] defines a neural network with flexible features that deals with the complexity of the tested time series for its prediction. In [164], the article talks of real world applications, by presenting a locally recurrent multilayer network to cope up with the complexity of the time series and to improve performance of the neural network model. The Radial Basis Function Network (RBFN) in [165] has been exploited to construct an adaptive fuzzy system based identification and prediction, whereas in [166], Leung has used the RBFN for the implementation of an optimal single-step ahead predictor for chaotic time series. Many other neural network architectures have been exploited in the framework of nonlinear system prediction: for example SOM [104]. However, there has been a little effort in implying Feed-forward topologies for time series prediction. As an example, the work in [167] talks of a local linear-modeling scheme based on SOM – Feed-forward topology. This scheme is composed of three steps of treatment. In the first step, the time series is embedded in space, the second step is performed by the SOM which associates input vector of the time series to a single local linear model. In the third step, the prediction is performed by selected local linear predictor.

We have already seen in Chapter 3 that research efforts combining machine learning techniques with CR technology do not provide any means of an efficient learning from past experiences, thus failing to exhibit one of the key properties of CR. Our work is different as we predict the spectrum evolution by exploiting the second order cyclostationary signal features detection, which is based on autocorrelation function, to construct a RF multivariate time series that contain more information than the univariate time series. The very challenging problem of Neural Networks based prediction of RF time series is approached in this Chapter by evaluating ERNN architecture in the paradigm of NARX time series.

4. Multivariate Time Series representation of received signal:

We have modelled the underlying process behind primary user feature detection leading to spectrum prediction, using time series, by defining a threshold and applying it to the consecutive observation results of the underlying process. We estimate the hypothesis shown in (6.1) by analyzing the received signal for peak value based on the previous observations and this estimation is compared to a predetermined threshold, which is again based on previous primary signal detection observations. If no periodicity is found, it means that there is no signal in the detected band. Otherwise, the band is used by primary user. The threshold is actually the optimized cost function of our designed ERNN. This primary user presence/absence predicate decision process is summarized in Figure 52. The effects of sampling scheme are out of scope of our work. Since ERNN well-represent the type of input/output relationship needed to model time series, they fit in our CR spectrum prediction scenario. We estimate the peak value at a given instance by modelling the previous observed values as RF NARX time series using ERNN.

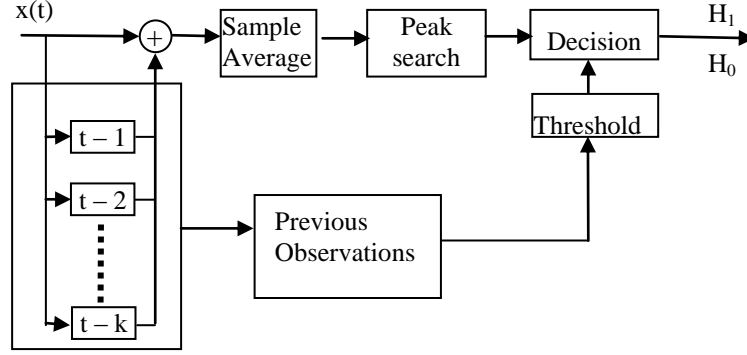


Fig. 52. Block Diagram of Spectrum Sensing predicate decision.

In this section, we formalize the RF time series in connection with the spectrum occupancy problem. The formulation of primary user presence prediction performed at t^{th} instance is given by,

$$H_t = H_1 / \mathcal{F}_{t-1} \quad (6.6)$$

\mathcal{F}_{t-1} is the generated NARX time series modelling sets regarding the RF data: cyclic prefix length, symbol period and preambles known from beginning to instance $t-1$, (where $N=t-1$). The correlation among these RF variables in our generated time series makes spectrum evolution prediction a difficult task. These modelling sets are formally represented as:

$$\begin{aligned} & x_1(1), x_1(2), x_1(3) \dots x_1(N); \\ & x_2(1), x_2(2), x_2(3) \dots x_2(N); \\ & \vdots \\ & x_n(1), x_n(2), x_n(3) \dots x_n(N). \end{aligned} \quad (6.7)$$

where, n is the number of variables considered in NARX time series and N is the length of input vector of the predictor p , which is actually the number of previous observations with respect to predicate H_t , shown in (6.1). This time series is generated using Exponential Moving Average Algorithm, used in several modern wireless applications [120]. Each member of the data sets shown in (6.7) differs with its adjacent member with a time delay of τ as each data set represents a single variable of cyclostationary process whose mean and autocorrelation exhibit periodicity.

We estimate the primary user presence/absence predicate at instance t , based on previous N observations. Assuming, for the moment, that we are using linear regression, we have the following formal representation,

$$E[H_t / \mathcal{F}_{t-1}] = \sum_{k=1}^N \beta_k H_{t-k} \quad (6.8)$$

E is the expected operator showing that the mean function of $x(t)$ is also periodical leading to the

characteristic of cyclostationary. β_k denotes the model parameters that control the associated RF features effect on the predicted value. Having the past N observations in memory, our proposed CR strives to obtain a time series model which represents the previous occupancy status and allows one to establish near field predictions. This leads to the prediction results based on knowledge of previous RF data relevant to the licensed user. However, since H_t is a binary predicate, we can not apply any linear model directly to equation (6.8) as the expected value for a binary process is in the interval $[0,1]$. To overcome this limitation, we apply a sigmoid link function for appropriate transformation. Sigmoid function provides a bounded output regardless of its input and is differentiable and real-valued, which is the reason to prefer it over other available link functions [168]. After applying the Sigmoid (tansig) function on our generated NARX time series set, our hypothesis definition becomes,

$$H_t = \frac{1}{1 + e^{-x(t)}} \quad (6.9)$$

The L.H.S. of Equation (6.9) can be expressed in following two ways:

i. Single Step ahead prediction:

$$H_t = H_1 / H_{t-1}, H_{t-2}, H_{t-3}, \dots, H_{t-N} \quad (6.10)$$

ii. Multi (n) Step ahead prediction:

$$H_{t+n} = H_1 / H_{t+n-1}, H_{t+n-2}, H_{t+n-3}, \dots, H_{t+n-N} \quad (6.11)$$

Where H_{t+n} is the $(t+n)^{th}$ time series estimated value, e.g. for $n=0$, equation (6.11) changes to single step ahead prediction case, shown in equation (6.10), which is the focus in this work. $x(t)$ in equation (6.9), is autoregression, defined in [138] as,

$$x(t) = a_0 + \sum_{k=1}^N (A_k H_{t-k}) + n(t) \quad (6.12)$$

where A_k is the coefficient matrices, $n(t)$ is the error vector, N is the model order, indicating the number of previous data points used for modelling and a_0 is the intercept. We predict the cyclic prefix length at time instance t , using all the generated RF modelling sets until instance $t-1$. We have already elucidated the suitability of this time series model construction to our problem in Section 2.4 of this Chapter.

5. ERNN Implementation:

In this section, we explore and justify our Neural Network choice. ERNN, despite being computationally expensive, well-project the dynamic properties needed to predict the chaotic time series. We have already explained in Section 4 the need of Sigmoid function for appropriate transformation. ERNN has inherent Sigmoid (tansig) neurons in its hidden layer and thus it inherently suits out problem of primary signal cyclostationary modelling using multi-variate time series.

Furthermore, they overcome the problem of attaining global minima in the most efficient way. If $E_s(t)$ is optimization objective, the aim is to train the NN to minimize the $E_s(t)$ by adapting the weight of the network. If S is the number of training samples of the net, then mathematically, the minimization objective is,

$$E_s(t) = \frac{1}{2} \sum_{t=1}^S \left\| \bar{Y}(t) - \hat{Y}(t) \right\|^2 \quad (6.13)$$

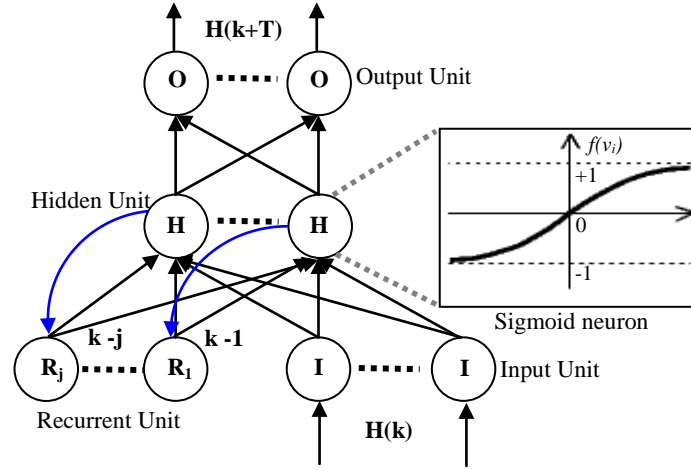


Fig. 53. Elman Recurrent Neural Network.

$\bar{Y}(t)$ and $\hat{Y}(t)$ are the observed value and the predicted value of the neural network, respectively. We have to train the neural network iteratively until $E_s(t)$ tends to be stable, with an acceptable value.

We have visualized our implemented ERNN in Figure 53, composed of three-layers with feedback from the hidden layer output to the selected neurons of first layer input. R_j is the j^{th} recurrent neuron that causes the delay for j time steps, thereby taking into account the previous j Spectrum occupancy observations. Sigmoid neurons in hidden layer are also highlighted. The output layer of our ERNN contains purelin neurons.

It is obvious from Figure 53 that the ERNN is different from traditional three-layer networks, as the hidden layer has a recurrent link with defined delays. This defined delay in the said connection is used to save the previous time instance values, which are exploited in the current time step. Furthermore, they are saved for the future time instances because of these recurrent links. The input to the network is our generated RF NARX time series modelling sets up to instance $t-1$. The output is the cyclic prefix length at instance t that decides for predicate H_t . This prediction based on previous observations leads to an intelligent cognitive user decision to exploit expected spectrum opportunities, thereby leading to optimized spectrum usage and interference avoidance.

At a specific (till a single instance before prediction) time instance ($t-1$), the previous activation of the hidden layer (until time $t-2$) and current inputs (at time $t-1$) are used as inputs to the network. At this stage, the network starts acting as a feed-forward topology by propagating these inputs forward for processing to produce the output. Once this step has been started, the activations of the hidden layer at time ($t-1$) are sent back through the recurrent links to the recurrent layer and the information is stored for the next training step t . The external input to the network at time ($t-1$) is $H(t-1)$ and the network output is $H(k+T)$, where $k=(t-1)$ and T decides the prediction steps ($T=1$ in our case). The inputs and outputs of different layer neurons of an ERNN in terms of k are explained hereafter: The input to i^{th} hidden neuron is $v_i(k)$, which is defined in equation (6.14).

$$v_i(k) = \sum_{j=1}^n w_{i,j}^P(k-1)P_j^R(k) + w_i^{H(k)}(k-1)H(k) \quad (6.14)$$

The output of j^{th} recurrent neuron is $P_j^R(k)$ as defined in equation (6.15).

$$P_j^R(k) = P_j(k-j) \quad (6.15)$$

The output of i^{th} hidden neuron is defined in equation (6.16).

$$P_i(k) = f(v_i) \quad (6.16)$$

The output of the net is given in equation (6.17).

$$H_k(k+T) = \sum_{i=1}^s w_i^{H_k(k+T)}(k-1)P_i(k) \quad (6.17)$$

If $i, j=1, 2, \dots, n$, then the link weightings between different layers is given by: $w_i^{H(k)}(.)$: input neurons and hidden neurons, $w_{i,j}^P(.)$: recurrent neurons and the hidden neurons and $w_i^{H_k(k+T)}$: hidden neurons and the output neurons. The real valued and differentiable Sigmoid activation function is denoted by $f(.)$. The objective of our designed ERNN is to minimize the squared error function shown in equation (6.13), to make the prediction more accurate. This Spectrum Evolution prediction method is evaluated by the criteria of Root-Mean-Square Error, E_{RMSE} , as shown in equation (6.18). This evaluation criterion is used to determine the acceptable threshold value, explained in equation (6.13).

$$E_{RMSE} = \left(\frac{1}{S-1} \sum_{t=1}^n [P_i - O_i]^2 \right)^{\frac{1}{2}} \quad (6.18)$$

E_{RMSE} reflects the absolute deviation between the predicted value and the target value. In equation (6.18) O_i is the target value and P_i is the predicted value. For impartial evaluation of the prediction performance of our approach, RF data for different time instances are used in the *training* (modelling) and *testing* (prediction) phases. The training of ERNN is recognized as a very delicate procedure and hence in the training phase, we should choose a learning algorithm that could facilitate us with this

task. The Numerical Methods based learning method, Levenberg-Marquardt (LM) training algorithm [169,170] is known to calculate the exact approximations by extrapolating the Hessian matrix. Hence, we have opted this learning method to train our NARX time series based ERNN. We explain briefly this training algorithm, allowing single step ahead prediction.

Equation (6.13) shows that the problem to be minimized is nonlinear least square minimization, provided that $S > \text{the dimension of time series}$. With the knowledge of numerical methods, we know that simple gradient descent and Gauss-Newton iteration are complementary in the advantages they provide. LM algorithm is based on this observation. The first derivatives of E_s , our minimization objective, can be written using the Jacobian matrix J of Y , defined as

$$J(t) = \frac{\partial Y_j}{\partial t_i}, 1 \leq j \leq S, 1 \leq t \leq S \quad (6.19)$$

The distinctive property of least-squares problem is that given the Jacobian matrix J , we can avoid the complexity of calculating Hessian matrix and it can be estimated as

$$H = J^T J \quad (6.20)$$

The gradient is computed as

$$g = J^T e \quad (6.21)$$

where e is the vector of network errors. Thus, the LM algorithm uses the approximations of equations (6.20) and (6.21) to calculate the next step weight update as

$$x_k = x_{k-1} - [J^T J + \mu I]^{-1} J^T e \quad (6.22)$$

We depict the two phases (training and testing) of our designed ERNN based spectrum prediction algorithm to process the cyclostationary RF time series data in Figure 54. This algorithm is designed for single step prediction, explained in equation (6.10).

6. Experimental Results:

This section analyses the experimental results obtained by our approach explained in Sections 4 and 5. As an example of licensed spectrum user, we have considered the UMTS frequency band whose usage is only 11.1% in Paris [136]. However, all other deterministic primary user signals can be exploited in the same way due to cyclostationarity. After generating the UMTS based RF time series modelling sets using Exponential Moving Average Algorithm, we train the first 300 state points of the 330 state points. The remaining 30 state points are taken as testing samples. The implementation parameters for our designed algorithm are listed in Table 17.

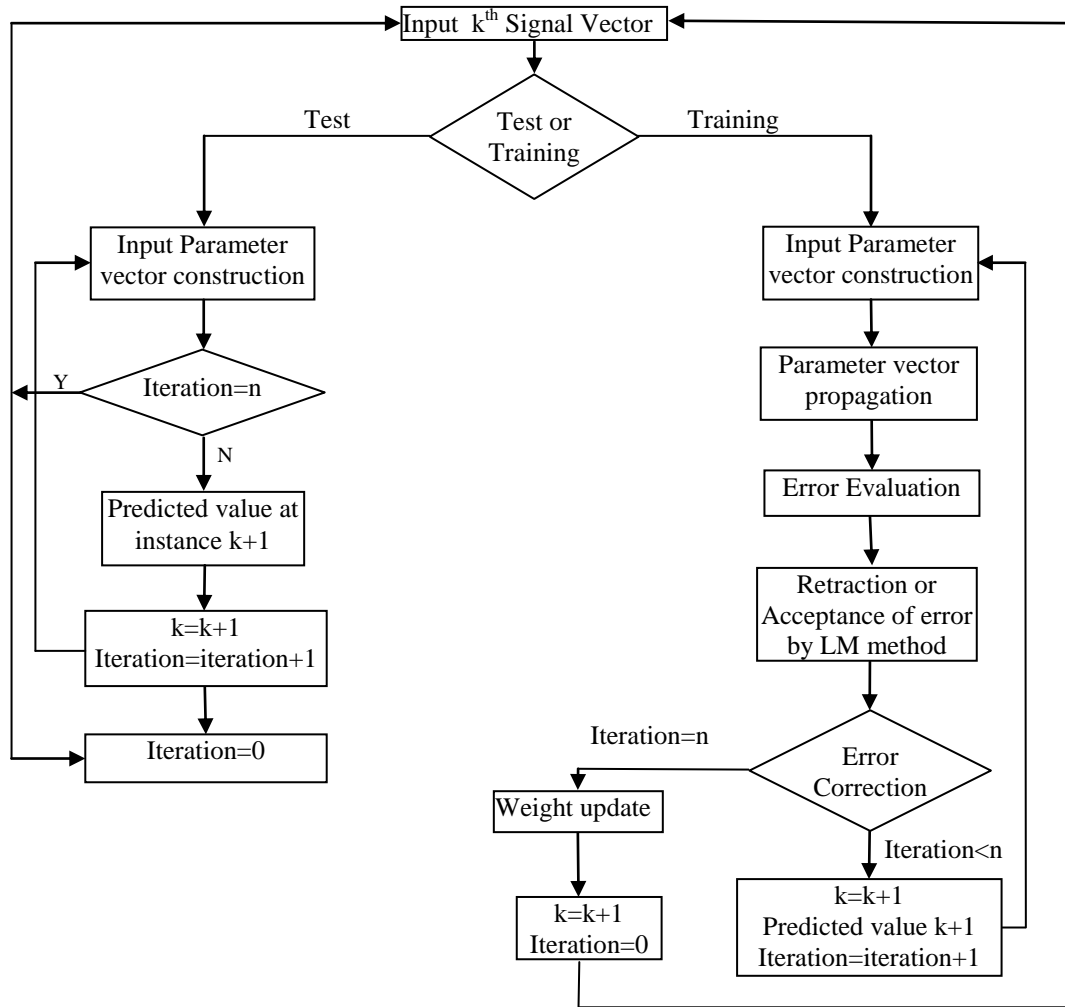


Fig. 54. Our designed ERNN and LM based Algorithm for Spectrum prediction in form of time series.

Implementation Parameters	
Number of input layer neurons	15
Number of hidden layer neurons	10
Number of output layer neurons	1
Training samples	300
Testing samples	30
Prediction steps	1
Variables in UMTS RF time series	3
Input dimension of Neural Network	30
Transfer Function	Sigmoid
Training Algorithm	LM

Tab. 17. Algorithm Parameters.

We have chosen these parameters after a careful analysis using trial and error, e.g. we analysed that hidden layer neurons greater than 10 lead to undesirable situation of overfitting. Using fewer NN parameters, than mentioned in Table 17, resulted in inadequate modelling, whereas more input

parameters excessively complicated our model, reminding us the computational complexity of ERNN.

The input to the ERNN, in both testing and learning phase, is trivariate RF time series formulated in Section 4. In each testing point, the input to NN consists of the three RF variables based time series of previous 30 time instances before time t , i.e. from instances $(t-30)$ to $(t-1)$. These instances are actually short duration observations. Mathematically,

$$\{x_1(1), \dots, x_1(N), x_2(1), \dots, x_2(N), x_3(1), \dots, x_3(N)\} = \{\vec{X_1^T}, \vec{X_2^T}, \vec{X_3^T}\} \quad (6.23)$$

The neural network's output is the predicted cyclic prefix length time series value at the instance t , thus deciding for the presence/absence of primary user signal based on previous 30 observations. We compare this predicted time series with the actual observed time series of our generated trivariate time series set to determine the prediction error that tells about our designed algorithm performance. We generate the initial weights randomly.

The primary user presence/absence prediction performance for testing samples is shown in Figures 55 and 56. In Figure 55 the red solid curve represents the predicted values during the 30 testing samples, whereas the dashed blue curve represents the observed values. The error between the predicted and observed values, called the prediction error, is shown in Figure 56. These Figures clearly demonstrate that the prediction is accurate as the predicted values have a similar trend as the observed values, leading to a slight prediction error. This prediction accuracy is attributed to our chosen Non-linear Autoregressive Exogenous Time Series Model, as it contains three times as many observations as are available for each independent modelling.

The E_{RMSE} is 0.0416 which outperforms the counterpart NN Techniques used for cognition incorporation in radios. Thus the primary user presence or absence predicate is predicted in the most efficient way. However, this efficiency is achieved at the cost of more complicated algorithm design because of the following two facts:

- The three considered RF features of the signal (variables of our trivariate time series) modelled by ERNN are interdependent.
- The inherent design complexity of ERNN.

The CR, thus, based on the previous observed features of the primary signal effectively predicts the presence or absence of primary user at a given near future instance. This leads to an efficient decision of cognitive user in terms of occupying or vacating a specific spectrum chunk at a specific time instance, thereby optimizing spectrum, the most valuable radio resource, usage.

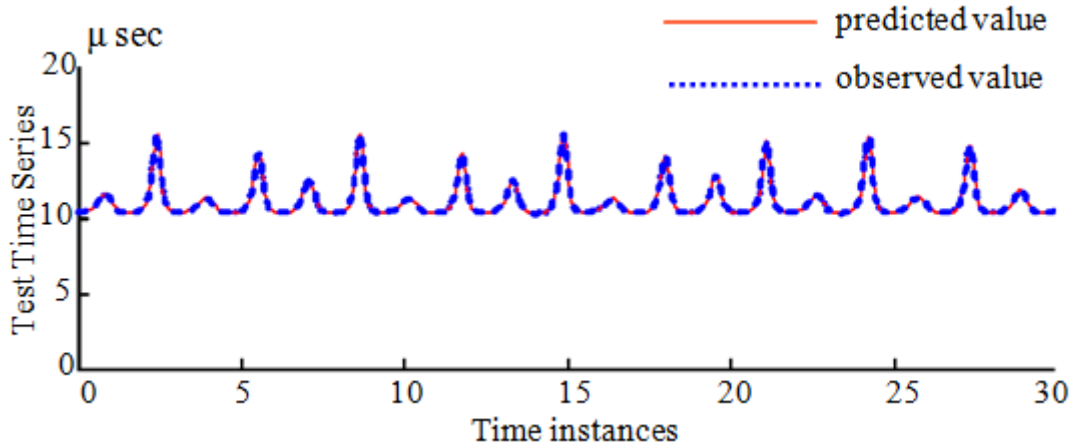


Fig. 55. Observed and Predicted values of time series over testing samples.

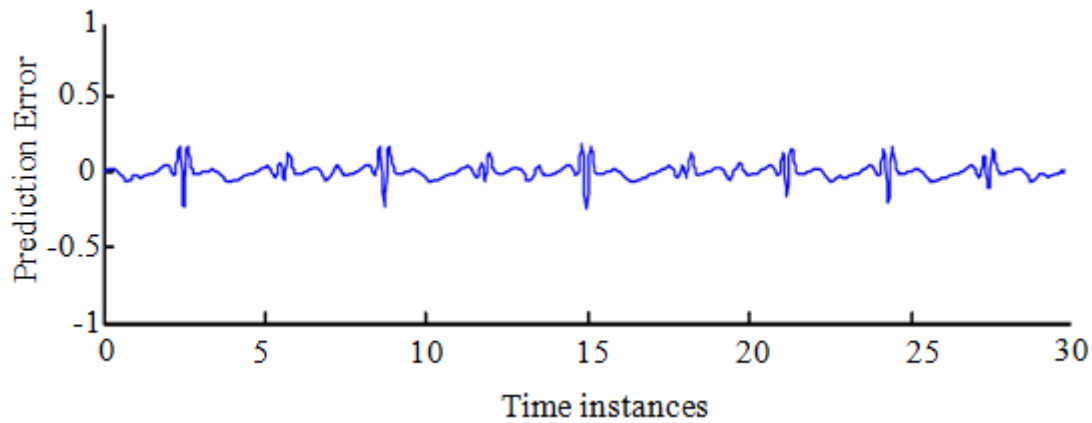


Fig. 56. Prediction error between predicted and observed time series.

7. Comparison with other Prediction Techniques:

This Section aims at comparing our experimental results shown in Figures 55 and 56 with that of other available prediction techniques available in literature. We have already cited a number of techniques, with an emphasis on ANN, that have been employed to predict time series in Section 3 of this Chapter. This Section compares different proposals for spectrum prediction, available in literature. The notion of predicting spectrum access was introduced by Clancy in [171]. He very briefly examined the hidden markov models for spectrum prediction. Wang [172,173] applies statistical method of ARIMA for time series prediction to model the GSM radio spectrum occupancy. Akaike's Information Corrected Criterion (AIC) was used to compare the performance with ARMA method to conclude that ARIMA gives better prediction results than ARMA. Figure 4 of [172] shows that the observed and predicted values are in good agreement, but the prediction is less accurate than our proposed neural networks

based method. Also, the statistical methods of ARMA and ARIMA have a very serious restriction with respect to CR that they require a priori knowledge of the underlying distributions of the observed process. The non-linear processes are known to give inaccurate prediction results when modeled using ARIMA [141]. It is important to mention that it has been proven experimentally that ANN are the most efficient techniques for prediction, e.g. in [174] authors group the statistical techniques and compare them with neural networks based prediction methods only to conclude that neural networks based predictors give better prediction performance than their statistical counterparts. Chen [175] proposes a three-order Hidden Markov Model to predict the spectrum by employing Wi-Fi as the primary user. The average error rate is used as performance measuring criteria to show that the complexity and memory space required for the higher order hidden markov model increases sharply with an increase in the number of orders. This is again in contrast to our ERNN based prediction method. Another hidden markov model based channel status predictor was proposed in [176] that is impractical for real scenarios as it only addressed deterministic traffic scenario. The details of model such as transition and output probabilities and number of states, in the case of hidden markov models are not provided, thus making it difficult to numerically compare it with our ERNN based model. However, it is an established fact [178] that neural networks offer better prediction performance than markov models and also overcome the huge memory space requirement. Furthermore, the continuous parameter update in hidden markov models is replaced by training in neural networks, which is not a continuous process, making it a pragmatic choice in the CR real time scenario. Yarkan [177] predicts the spectrum using binary time series; however, the results are degraded as the problem of attaining global minima is not addressed. The neural networks based predictors are known to address this problem, which is the reason of our superior prediction results as compared to [177]. The prediction efforts using neural networks, in the context of cognitive radios to predict data rate [120] and throughput [122] do not use the time series modeling. This can be attributed to the error values of 0.0637 and 0.8 respectively, again proving that our numerical result value of 0.0416 is better. In a similar effort to ours, Tumuluru [178] uses MLP neural networks to predict future expected spectrum holes and claims that 60% improvement in spectrum utilization is achieved by MLP neural networks based prediction methods. However, MLP is a feed forward topology and hence feedback connections extending from output of neurons to previous layer input neurons are absent. In our prediction context, we need to preserve the history of previous RF features and hence, we need the feedback links. This explains our ERNN choice as compared to MLP.

We have observed that different spectrum prediction techniques have different error evaluation criteria pertinent to that specific technique. However, we observed the specific characteristics and prediction results of the different techniques to come up with the comparison done in this Section.

8. Conclusions:

Today, spectrum policy is undergoing radical rethinking, which motivates us to seek a greater cross-domain interaction. In this respect, we proposed a novel approach to solve the spectrum scarcity paradox. We designed an algorithm, using Nonlinear Autoregressive Exogenous Model that exploits the inherent cyclostationary features of a primary user signal to predict the future spectrum evolution. Our ERNN based algorithm accepts trivariate UMTS RF features time series to predict the future presence/absence of primary user. The obtained results are very promising and clearly demonstrate the superior performance of the proposed approach that gives a very slight prediction error, thus outperforming other NN based cognition incorporation solutions. The successful primary user presence/absence prediction enables CR to exploit available spectrum opportunities. This leads to optimized spectrum usage.

Chapter 7:

Final Word and Perspectives.

Lateral thinking is perhaps the most important state of mind one can possess, because it can bring a brand new start to a specific domain, simply by watching and observing the phenomena and reasoning from other disciplines and creating a link by a tighter coupling with our domain. As an example, lateral thinking has helped in developing several meta-heuristics which are inspired from nature, biology, economics, physics, engineering or different disciplines. There are several phenomena that we don't understand in nature, but which are already working efficiently. This thesis emphasized on gaining a greater cross-domain interaction by getting inspiration from natural systems, thereby answering the ITRS Roadmap Prediction. We combined artificial intelligence, parallel processing, statistics, and physics to engineer a Resource Efficient Universal Multi-standard Transceiver Software Defined Radio Waveform. The main focus of this thesis was to answer the different adaptability and reconfigurability aspects required to realize the SDR and CR technology. Our designed waveform efficiently addresses the following three most important aspects of SDR and CR:

1. The inconsistency between hardware and software advancements, thereby preventing to push the communications hardware into obsolescence in upcoming years, by designing an efficient parallelization technique for each of the SDR waveform component.
2. A single Multi-standard adaptive Transceiver capable of detecting multiple communication standards, based on three ANN techniques and switches to appropriate ANN for appropriate standard detection and operation.
3. A spectrum reuse opportunity in underutilized radio spectrum is predicted, based on previous cyclostationary features of primary user, thereby giving the secondary user an opportunity to exploit the spectrum holes in the near future. ANN is used for prediction of cyclostationary RF features, modeled by time series, of primary signal.

The above three aspects of our waveform can lead to the development of smarter devices that are capable of adaptability requirements of SDR and CR embedded implementation and that these agile frequency requiring devices may be seen as an evolution towards intelligent and high speed ubiquitous wireless communications that uses and re-uses radio resources effectively. We have justified by

proven evidences, observations and logical deductions the above mentioned three aspects of our designed waveform. After reviewing the important research projects and machine learning techniques that can be applied to this domain, the first portion of the thesis showed that by shifting short term development of a radio waveform into software domain, we can benefit from the capabilities of multiprocessor platforms to enhance the performance of the computationally intensive SDR waveform components. We contributed to better parallel programming techniques through accurate and real time NoC monitoring. We further optimized the resource utilization by partitioning the PEs for different waveform components and thus further optimized the multicore chip utilization. It remains that despite considerable research, there are very few existing implementations of actual MPSoC. There is a need to design and build MPSoC platforms with a significant number of processors as experimental tool to feedback the target MPSoC for SDR waveform.

It can be interesting, in future, to exploit Moore's Law by engineering a CORBA based distributed SDR waveform comprising of the same resources as addressed in this thesis to analyze the CORBA delays in further detail. In this respect, we have already provided a performance analysis by collecting performance metrics on modern high speed network of multi-core workstations using OmniORB CORBA implementation as the target ORB. We measured the five aspects namely, Invocation Time, In Sequence, Out Sequence, Objects Registered and Multithreading. Since modern high speed multi-core systems and efficient light weight and modern implementations of CORBA such as omniORB have made it possible to address the issues of CORBA delays in a Software Defined Radio, in case if the resources of a waveform are in different address spaces, therefore our benchmarking results can serve for an efficient SDR distributed waveform, which is a possible future research direction.

The highly varying environments characterizing the future of wireless communications is the focus on the second part of this thesis. We exploit the inherent parallelism in three ANNs: LVQ, SOM and MLP, for answering the high adaptability requirements of CR embedded implementation by mapping the three algorithms on the aforementioned MPSoC. These three algorithms constitute the second aspect of the thesis. We propose a multistandard CR Transceiver based on these three ANN techniques, which are widely different from conventional information processing as they have the ability to learn from previous examples, thus exhibit the most basic instinct of CR. We exploit the inherent property of SOM, Tonotopy, to recognize the standard in our proposed multi-standard CR Transceiver. In the case of MLP, we have used Mean Square Error as a metric for measuring MLP's performance. The ANN based algorithms, apart from possessing the basic instinct of CR, *learning*, are inherently parallel in nature. We exploited this inherent parallelism. Our designed synchronization and parallelization strategy for the Multi-standard Universal Transceiver gave excellent speed-up. An efficient load-balancing across all the 16 PEs is done in the case of a received signal, which may be the result of summation of many standards. However, it appears to us that something important is missing in our understanding of the electromagnetic waves which has led us to build "dumb

transceivers". In this respect, this contribution is a first step in the expected technological breakthrough that will eventually replace the dumb transceivers with smarter cognitive multi-standard transceivers.

All the wireless applications use spectrum to work and are subjected to spectrum reuse constraints to avoid radio interference. Our third contribution is towards the efficient spectrum management, a concept that is often seen as the defining notion for CR. We solved the spectrum scarcity paradox by Nonlinear Autoregressive Exogenous Time Series Modeling of UMTS RF features. This time series is given as input to ERNN that predicts the future spectrum evolution in time, thus allowing cognitive user to exploit the available spectrum opportunity. Error evaluation criteria Root-Mean-Square Error for ERNN is used. We observed a similar trend between predicted and observed values, giving us a slight prediction error. So, in this contribution, we proposed an ERNN based solution to detect the future possible spectrum holes. This intelligent algorithm that accepts RF multivariate time series as input makes the best use of a spectrum part, which leads to a new way of spectrum allocation and exploitation. The cognitive radio concept is often seen as the solution for the spectrum scarcity problem, however, we have shown that there are hurdles to overcome on the way towards fully flexible cognitive radios. We have addressed these hurdles that include mainly hardware constraints that result from the need for high degree of reconfigurability and adaptability to any given air interface. We have also presented and discussed the currently ongoing research activities related to Software Defined Radio and Cognitive Radio. High performance SDR embedded applications based on single processing will increasingly be moved to embedded multiprocessors. In this respect, combining all the three aspects, this doctoral thesis has introduced solutions to the most important questions related to SDR and CR engineering. As discussed previously, the main advantage of the proposed algorithms for Spectrum efficient Universal Transceiver possessing SDR waveform is high degree of reconfigurability and adaptability. Also, the general purpose embedded solution has become the standard for implementing high adaptability requiring devices as compared to their counterpart application specific solutions. Thus this doctoral thesis shows the adaptability to varying situations and unknown systems illustrating the value of SDR and CR algorithms suitability on general purpose hardware against the better performing but less flexible dedicated hardware.

As ANN is a meta-heuristic approach, it can be used to solve the other problems from cognitive world, e.g. a possible future access of research to contribute to our third contribution could be the exploration of handover strategies between different Standards. We can extend our work by constructing another multivariate time series based on the features of WiMAX and then use the ANN to explore the inter-standard handover strategies. This will enable to have an acting mobile terminal for WiMAX and UMTS, at the same time. The intersystem handover performance using ANN can be analyzed to explore the mobility of the two networks. However, for this task we need to design a specialized ANN that has the qualities of SOM to recognize a standard on the basis of input parameters followed by the

recurrent features of the ERNN to model the multivariate time series of the respective norm. The usage of real data instead of generated data for time can be another improvement.

Thus, there are multiple future directions that can be pursued in the framework of the problems addressed in this thesis.

Publications

- M.I.Taj and M.Akil, “Cognitive Radio Multiple-step ahead Spectrum Evolution Prediction using Artificial Neural Networks based Multivariate Time Series Modelling.” *Journal of Signal and Information Processing. Submitted.*
- M.I.Taj, and M.Akil, “Standard Recognizing Artificial Neural Networks Based Reconfigurable Cognitive Radio Transceiver and its Embedded Implementation.” *IET Journal: Special issue on Cognitive Communications. Under Review.*
- M.I.Taj and M.Akil, “Cognitive Radio Spectrum Evolution Prediction using Artificial Neural Networks based Multivariate Time Series Modelling.” The 17th IEEE European Wireless Conference, April, 2011. Vienna, Austria.
- M.I.Taj, M.Akil and O.Hammami ”Standard Recognizing Self Organizing Map based Cognitive Radio Transceiver”, The 5th IEEE International Conference on Cognitive Radio Oriented Wireless Networks and Communications, 2010, Cannes, France
- M.I.Taj, O.Hammami and M.Akil,”SDR waveform components implementation on single FPGA Multiprocessor Platform” The 17th IEEE International Conference on Electronics, Circuits, and Systems, 2010. Greece.
- M.I.Taj, K. Huggins and O.Hammami, “OSSIE Signal Processing Functions Performance Enhancements through Parallelization in an embedded multiprocessor architecture” First Software Defined Radio Technical Conference and Product Exposition, Madrid, Spain, April,2009.
- **Best Paper Award** M.I.Taj, O.Hammami and K. Huggins, “Performance Evaluation of SDR on embedded platform: The case of OSSIE” The 2nd IEEE International Conference on Computer, Communications and Control, 2009. Pakistan.
- M.I.Taj and O.Hammami, “Simultaneous Power and Bit Allocation through Genetic Algorithm in a MB-OFDM System”, IEEE International Conference on Advanced Communication Technologies, February, 2008. South Korea.

References

- [1] www.wirelessinnovation.org
- [2] J Mitola, "The Software Radio," IEEE National Telesystems Conference, 1992 - Digital Object Identifier 10.1109/NTC.1992.267870
- [3] R.J.Lackey and D.W.Upmal, "SPEAKEasy, the Military Software", IEEE Communications Magazine, vol.33, No.5, May, 1995. Pp.56-61.
- [4] V.G. Bose, "Design and Implementation of Software Radios using a General Purpose processor", Ph.D. dissertation, MIT, 1999.
- [5] www.vanu.com
- [6] E. Buracchini, "The Software radio Concept", in IEEE Commun. Magazine, vol.38, No.9pp.138-143. September, 2000
- [7] OSSIE ,Open Source SCA Implementation ::Embedded, www.ossie.wireless.vt.edu/trac/
- [8] C.R.A.Gonzales, C.B.Dietrich, S.Sayed, H.I.Volos, J.D.Gaeddert, P.M.Robert, J.H.Reed and F.E.Kragh, "Open-Source SCA-Based Core Framework and rapid Development Tools Enable Software-Defined Radio Education and Research", IEEE Communications Magazine, October, 2009.
- [9] C.Lee, J.Kim, S.Hyeon and S.Choi, "FPGA Design to support a CORBA component", in SDR'08 Technical Conf. Product Exposition, October, 2008.
- [10] J.Glossner, M.Moudgill, D.Iancu, G.Nacer, S.Jinturkar, S.Stanley, M.samori, T.Raja, M.Schulte and S.Vassiliadis, "Future Wireless Convergence Platforms", Proceedings of the 3rd IEEE/ACM/IFIP international conference on Hardware/Software codesign and system synthesis, Pages: 7-12 Year of Publication: 2005 Jersey City, NJ, USA.
- [11] J.Glossner, D. Iancu, M.Moudgill, G.Nacer, S.Jinturkar, S.Stanley and M.Schulte, "The Sandbridge SB3011 Platform", EURASIP Journal on Embedded Systems, Volume 2007, Issue 1. January 2007.
- [12] Wireless Interoperability for Security (WINTSEC) <http://sta.jrc.ec.europa.eu>
- [13] EULER project <http://www.euler-project.eu/>
- [14] European approach for SDR certification », European defense Agency (EDA) presentation at the Wireless Innovation Forum Workshop, "SCA Test, Evaluation and Certification"Rome, April, 2004.
- [15] ETSI TR 102 680, "Reconfigurable Radio Systems (RSS); SDR Reference Architecture for Mobile Devices.", 2009
- [16] IEEE Standards Association. IEEE Standards Coordinating Committee 41 (Dynamic Spectrum Access networks). <http://www.scc41.org>
- [17] "Software Communications Architecture Specifications", 15 May, 2006 Version 2.2.2, JTRS Standard, Joint Program Executive Office of the Joint Tactical Radio System <http://jtrs.spawar.navy.mil/sca>
- [18] SCARI <http://www.crc.gc.ca/en/html/crc/home/research/satcom/rars/sdr/sdr>
- [19] www.spectrumsignal.com
- [20] T.Ulversoy, "Software defined Radio: Challenges and Opportunities",IEEE Communications Surveys and Tutorials, Vol.12, No.4, Fourth Quarter 2010.
- [21] Simulation and Software Radio for mobile communications : Edited by: H.Harada and R.Prasad. ISBN: 1-58053-044-3

- [22] J.Mitola III, "Cognitive Radio: An integrated agent architecture for software defined radio", Ph.D. Dissertation, Royal Institute of Technology, Stockholm, Sweden, May, 2000
- [23] S.Haykin, "Cognitive Radio: Brain-empowered wireless communications", IEEE J.Select. Areas Commun. Vol.23, no.2, pp.201-220, Feb.2005
- [24] S.Shantaraskul and K.Moessner, "Implementation of a generic algorithm based decision making framework for opportunistic radio", IET Communications. Pp. 495-506, ISSN: 1751-8626 2010.
- [25] J.Mitola III and G.Q.Maguire, Jr, "Cognitive Radio: Making software radios more personal", IEEE Personal Commun. Mag. Vol.6, no.4, pp.13-18, August 1999.
- [26] A.Amanna, J.H. Reed, "Survey of Cognitive Radio Architectures", Proceedings of the IEEE SoutheastCon 2010 SoutheastCon (2010).
- [27] G.K.Rauwerda, G.J.M.Smit, L.F.W, van Hoesel, P.M.Heysters, "Mapping Wireless Communication Algorithms to a Reconfigurable Architecture", Proceedings of ERSA'03, Las Vegas, USA. pp. 242-251. ISBN 1-932415-05-X, July, 2006.
- [28] A.B.MacKenzie, J.H.Reed, P.Athanas, C.W.Bostian, M.Buehrer, L.A.DaSilva, S.W.Ellingson, Y.Thomas Hou, M.Hsiao, J.M.Park, C.Patterson, S.Raman, C.R.C.M. DaSilva, "Cognitive Radio and Networking Research at Virginia Tech", Proceedings of the IEEE. Vol.97, No.4, April 2009.
- [29] R. Chen, J.M.Park and J.H. Reed, "Defense against primary user emulation attacks in cognitive radio networks", IEEE J. Select. Areas Commun., Vol.26, no.1, pp.25-37, January, 2008.
- [30] J.Mitola III. "Cognitive Radio for flexible mobile multimedia communications", 6th IEEE International Workshop on Mobile Multimedia Communications, San Diego , CA 15-17 Nov.1999. pp.3-10
- [31] J.Mitola III, "Cognitive Radio Architecture Evolution", Proceedings of the IEEE Vol.97, No.4, April, 2009
- [32] H.Huang, Z.Zhang, P.Cheng, G.Yu and O.Qui, "Throughput analysis of cognitive MIMO system", in Proc. Int. Workshop Cross Layer Design (IWCLD'07), Jinan, China, September, 2007
- [33] D.Novo, B.Bougard, P.raghavan, T.Schuster, L.Van der Perre (IMEC, Leuven, Belgium), and H.S. Kim and H.Yang (Samsang).2006, "Energy Performance Exploration of a CGA based SDR processor", in Proc. SDR Forum Tech Symp.
- [34] M.A. Beach, C.M.Tan and A.R.Nix, "Indoor dynamic double directional measurements", in Proc. Int Conf. Electromagn. Adv.Applicat, Turin, Italy, 2003
- [35] C.Clancy, J.Hecker, E.tuntebeck, T.O'Shea, "Applications of Machine Learning to Cognitive Radio Networks", IEEE Wireless Communications, August, 2007.
- [36] M.I.Taj and O.Hammami, "Simultaneous Power and Bit Allocation through Genetic Algorithm in a MB-OFDM System", IEEE ICAC, February, 2008. South Korea.
- [37] D.Nussbaum, K.Kalfallah, R.knopp, C.Moy, A.Nafkha, P.Leray, J.Delorme, J.Palicot, J.Martin, F.Clermidy, B.Mercier, R.Pacalet, "Open Platform for prototyping of Advanced Software Defined Radio and Cognitive Radio techniques", 12th Euromicro Conference on Digital System Design, Architectures, Methods and Tools. 2009. 978-0-7695-3782-5/09
- [38] Technical details of Project "Gestion du spectre et Radio Cognitive" submitted to RNRT, France. 2005. <http://www.eurecom.fr/~haddadm/>
- [39] M.Muck,D.Bourse,K.Moessner,N.Alonistioti,P.Demestichas,E.Nicollet,E.Buracchini,D. Bateman, Z. Boufidis, E. Patouni, V. Stavroulaki, A. Trogolo, and P. Gorla, "End-to-End Reconfigurability in Heterogeneous Wireless Systems – Software and Cognitive Radio Solutions enriched by Policy- and Context-based Decision Making," in Proc.16th IST Mobile and Wireless Communications Summit, pp. 1–5,2007.

- [40] J.Kumagai, "Winner : Radio Revolutionaries", IEEE Spectrum. January, 2007.
- [41] J.H. Reed, "Software Radio: A Modern Approach to Radio Engineering, Publisher: Prentice Hall, ISBN-13:9780130811585.
- [42] M.Cummings and S.Haruyama, "FPGA in the Software Radio", in IEEE Communication Magazine, 1999, Vol.37, pp.108-112.
- [43] J.Fifield,"A software defined OFDM Modulator", Master's Thesis, University of Colorado, Boulder, 2006.
- [44] X.Revés, V.Marojevic, R.Ferrus, A.Gelonch, "FPGA's Middleware for Software Defined Radio Applications", International Conference on Field Programmable Logic and Applications, Aug. 2005. Pages: 598-601.
- [45] G.J. Minden, J.B. Evans, L.Searl, D.DePardo, V.R.Petty, R.Rajbanshi, T.Newman, Q.Chen, F.Weilding, J.Guffey, D.Datla, B.Barker, M.Peck, P.Cordill, A.M. Wyglinski and A.Agah, "KUAR:A Flexible Software Defined Radio Development Platform".
- [46] G.Schelle, J.Fiffield and D.Grunwald, "A Software Defined Radio Application utilizing modern FPGAs and NoC Interconnects", International conference on Field Programmable Logic and Applications, August 2007. FPL, 2007.
- [47] S.Dikmese, A.Kavak, S.Sahin, K.Kucuk and H.Dincer, "Evaluation of FPGA based Software Radio Beamformers for 3G Wireless ", IEEE Radio and Wireless Symposium, 2007.
- [48] M.Awan, M.M. Alam, P.Koch and N. Behjou, "Design and implementation of an FPGA-based Multistandard Software Radio receiver", Norchip, 2007, Issue, 19-20 Nov.2007 Page(s) 1-5.
- [49] H.R. Myler, S.A.Bagasrawala, N.V.Narayana, "A concurrent processing approach for Software Defined Radio Baseband Design", IEEE Region 5and IEEE Denver Section Technical, Professional and Student development Workshop. 2005.
- [50] R.C. Reinhart, M.C.Scardelletti and D.J. Mortensen, "Hardware Architecture Study for NASA's Space Software Defined Radios", IEEE Annual Wireless and Microwave Technology Conference, 4-5 December, 2006.
- [51] Y.Tachwali, H.Refai, "Implementation of a BPSK Transceiver on Hybrid Software Defined Radio Platforms", Third International conference on Information and Communication Technologies: from theory to applications, 2008.
- [52] M.Rakhshandar, M.Teimouri, and Z. Hassan Shahi, "4th IEEE International Conference on Implementation of Software Radio based on PC and FPGA", Circuits and Systems for Communications, May, 2008, ICCSC 2008, Pages: 633-637. Location: Shenghai.
- [53] G.Abgrall, F.L.Roy, J.P.Delahaye, J.P.Diguet, G.Gogniat, "A Comparative Study of two Software defined Radio Platforms", SDR Technical Conference, Oct.26-30, 2008. Location: Washington DC.
- [54] Y.Lin, H.Lee, M.Woh, Y.Harel, S.Mahlke and T.Mudge,"SODA: A Low Power Architecture for Software Radio", Proceedings of the 33rd International Symposium on Computer Architecture, Pages: 89-101. Year of Publication, 2006.
- [55] B. Bougard, B.D. Sutter, S.Rabou, D.Novo, O.Allam, S.Dupont, L.V. der Perre, " A Coarse-Grained Array based Baseband Processor for 100 Mbps+Software Defined Radio", Design, Automation and test in Europe, March 10-14, 2008. Munich, Germany.
- [56] A.C.H. Ng, J.W.Weijers, M.Glassee, T.Schuster, B.Bougard, L.V. der Perre, " ESL Design and HW/SW Co-verification of High-end Software Defined Radio Platforms", Proceedings of the 5th IEEE/ACM International Conference on Hardware/Software codesign and system synthesis. Pages: 191-196, Year of Publication: 2007, Salzburg, Austria.
- [57] H.Lee and T. Mudge , " A Dual-Processor solution for the MAC Layer of a Software Defined Radio Terminal", Proceedings of the 2005 International Conference on Compilers, Architectures

- and Synthesis for embedded systems, pages: 257-265, Year of Publication:2005, San Francisco, California, USA.
- [58] Y.Lin, M.Mahlke and T. Mudge, “Hierarchical Coarse-grained Compilation for Software Defined radio”, Proceedings of the 2007 International Conference on Compilers, Architectures and synthesis for Embedded Systems, Pages:115-124. Year of Publication 2007, Salzburg, Austria.
 - [59] R.Schiphorst, F.W. Hoeksema and C.H. Slump, “ A Real-Time GPP Software Defined Radio Testbed for the Physical Layer of Wireless Standards”, EURASIP Journal on Applied Signal Processing, Volume 2005, Issue 1, Pages 2664-2672, Year of Publication: 2005 ISSN:1110-8657
 - [60] <http://www.itrs.net>
 - [61] Z.Wang, “Design and Multi-Technology Multi-objective Comparative Analysis of Families of MPSoC”, Thèse soutenue Novembre 2009 à l’ENSTA ParisTech, France.
 - [62] “ARM 11 MPCore, <http://www.arm.com/products/CPUs/ARM11MPCoreMultiprocessor.html>
 - [63] MIPS32 1004KTM Core, <http://www.mips.com/products/processors/32-64-bit-cores/mips32-1004k>
 - [64] S.Shibahara, M.Takada, T.Kamei, et al, “SH-X3:SuperH multicore for embedded systems”, in Proceedings of the 19th Symposium on High Performance Chips (Hot Chips ’07), USA, August 2007
 - [65] “Texas Instruments Multicore Fact Sheet SC-07175”.
 - [66] “Texas Instruments TMS320C6474 Multicore DSP SPRS552”, October, 2008
 - [67] “Texas Instruments TMS320VC5441 Fixed-Point DSP data manual SPRS122F”, October 2008
 - [68] “QORIQ™ P4080 Communications Processor”,
<http://www.freescale.com/webapp/sps/site/overview.jsp?nodeId=0162468rH3bTdG25E4>.
 - [69] T.Miyamori, ”Venezia : a scalable multicore subsystem for multimedia applications”, in Proceedings of the 8th International Forum on Application Specific Multiprocessor SoC, Germany, June 2008.
 - [70] Arteris Danube 1.12, Packet Transport Units technical reference, o4277v11, April, 2008.
 - [71] Microblaze processor reference guide. Xilinx user guide 081 (v.7.0).
 - [72] M.Rice, F.Harris, “Polyphase Filterbanks for Symbol Timing Synchronization in Sampled Data Receivers”, MILCOM , 2002 Proceedings. Publication Date: 7-10 October, 2002. Vol.2. Page(s):982-986 vol.2
 - [73] www.xilinx.com/ml403
 - [74] M.B.Kamble and K.Ghosse, “Analytical energy dissipation models for low power caches”, in proceedings of International Symposium on Low Power Electronics and Design”, Aug.1997, pp.143-148.
 - [75] S.Kannan and J.Fridman, “Cached Memory Performance Characterization of a Wireless Digital Baseband Processor”, International conference on Acoustics, Speech and Signal Processing, 2004. Proceedings 17-21 May, 2004, Austin, TX, USA.
 - [76] H.S.Kim, A.K.Somani and A.Tyagi, ”On Reconfigurable Cache for Computing”, Seventh Annual IEEE Symposium on Field-Programmable Custom Computing Machines, 1999. FCCM’99. proceedings.
 - [77] M.Palkovic, H.Cappelle, M.Bougard, L.Van der Perre, “Mapping of 40 MHz MIMO SDM-OFDM Baseband Processing on Multi-processor SDR Platform”, The 11th workshop on Design and Diagnostics of Electronics, Circuits and Systems. 16-18 April, 2008.

- [78] B.Meï, S.Vernalde, D.Verkest, H.De man, R.Lauwereins, "ADRES: an architecture with tightly coupled VLIW processor and coarse-grained configurable matrix", Proc IEEE Conf. on Field Programmable Logic and its Applications (FPL), Lisbon, Portugal. pp.61-70, Sep, 2003.
- [79] P.J.Balister, "A Software Defined Radio Implemented using the OSSIE Core Framework Deployed on a TI OMAP Processor", Masters Thesis submitted to Virginia Polytechnic Institute and State University, USA. December, 4, 2007.
- [80] A.N.Choudhary,S.Das, N. Ahuja and H. Patel,"A Reconfigurable and Hierarchical Parallel Processing Architecture: Performance Results for Stereo Vision" 10th International Conference on Pattern Recognition, 1990. Proceedings Volume ii, Issue , 16-21 Jun 1990 Page(s):389 - 393 vol.2
- [81] J.H. Bahn, J. Yang and N. Bagherzadeh,"Parallel FFT Algorithms on Network-on-Chips", Fifth International Conference on Information Technology: New Generations Pages.1087-1093. Year of publication:2008. ISBN:978-0-7965-3099-4
- [82] A.H.Kamalizad, C.Pan and N. Bagherzadeh,"Fast Parallel FFT on a Reconfigurable Computation Platform" 15th Symposium on Computer Architecture and High Performance Computing, 2003. Proceedings. Volume , Issue , 10-12 Nov. 2003 Page(s): 254 – 259
- [83] G.Zhong, F.Xu and A.N.Wilson,"A Power-Scalable Reconfigurable FFT/IFFT IC Based on a Multi-Processor Ring", IEEE Journal of Solid State Circuits, Vol. 41. No.2, February, 2006. Page(s):483-495, ISSN:0018-9200.
- [84] H.He and H.Guo, "The realization of FFT Algorithm based on FPGA Co-processor", Second International Symposium on Intelligent Information Technology Application, 2008. Vol.3. Page(s). 293-243
- [85] A.R. Rhiemeier, "Benefits and Limits of Parameterized Channel Coding for Software Radio", Proceedings of the 2nd Karlsruhe Workshop on Software Radios, Karlsruhe (Germany), March 20/21, 2002, ISSN 1616-6019, pp. 107 – 112
- [86] J.Palicot and C.Roland, "FFT: A Basic Function for a Reconfigurable Receiver", ICT'03 Conference, Tahiti, France, 2003.
- [87] C.Moy, J.Palicot, V.Rodriguez and D.Giri, "Optimal Determination of Common Operators for Multi-Standard Software Defined Radio", 4th Karlsruhe Workshop on Software Radios, Karlsruhe (Germany), 22-23 March, 2006.
- [88] S.T.Tabassum Gul, C.Moy, J.Palicot, "Two Scenarios of Flexible Multi-Standard Architecture Designs using a Multi-Granularity Exploration", The 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'07)
- [89] L. Alaus, J.Palicot, C.Roland, Y.Louët and D.Noguët, "Promising Technique of Parameterization For Reconfigurable Radio, the Common Operators Technique: Fundamentals and Examples", March 2009, Journal of Signal Processing Systems, Springer New York ISSN 1939-8018 (Print) 1939-8115 (Online), DOI 10.1007/s11265-009-0353-4.
- [90] L.Alaus, D.Noguët and J.Palicot, "A reconfigurable linear feedback shift register for software radio terminal", Santorin: ISWPC'08, 2008.
- [91] A.A Ghouwayel and Y. Louët "FPGA implementation of a re-configurable FFT for multistandard systems in software radio context", IEEE Trans. On Consumer Electronics Journal, vol. 55, n°2, pp. 950-958, May 2009.
- [92] M. Naoues, L. Alaus and D. Noguët, A common operator for FFT and Viterbi algorithms, 13th Euromicro Conference on Digital System Design, Lille, France, 2010.
- [93] H.Wang, Y.Louët, J.Palicot, L.Alaus and D.Noguët, "Memory-efficient FFT architecture using R-LFSR based CORDIC common operator", 2nd International Workshop on Cognitive Information Processing, 2010.

- [94] J. S. Reeve, K.Amarasingh "A parallel Viterbi decoder for Block Cyclic and Convolution codes" IEEE Signal Processing, 2005. Page(s) 273-278.
- [95] S.Nishijima, M.Saito and I.Sugiyama, "Single-Chip Baseband Signal Processor for Software-Defined Radio", Fujitsu scientific and technical journal, ISSN 0016-2523, vol.42, no.2, pp. 240-247, 2006.
- [96] Z.Miljanic, I.Seskar, K.Le and D.Raychaudhuri, "The WINLAB Network centric Cognitive Radio Hardware Platform – WiNC2R", Mobile Networks and Applications. Vol.13, Number 5, pp.533-541, 2008.
- [97] Q.Zhang, G.J.Smit, A.Kokkeler, F.W.Hoeksema and m.Heskamp, "A Reconfigurable Platform for Cognitive Radio", 2nd International Conference on Mobile Technology, Applications and Systems, 2005.
- [98] Q.Zhang, A.B.J.Kokkeler, G.J.M.Smit, K.H.G.Walters, "Cognitive Radio baseband processing on a reconfigurable platform", Volume 2, Pages 33-46 Physical Communications, 2009.
- [99] A.Wiesler and F.K.Jondral, "A software radio for second and third generation mobile systems", IEEE Transactions on Vehicular technology, vol.51, pages 738-748. 2002.
- [100] F.K.Jondral, A.Wiesler and R.Machauer, "A Software Defined Radio structure for 2nd and 3rd generation mobile communications standards", 6th IEEE International Symposium on Spread Spectrum Techniques and Applications, vol.2 pages. 637-640, September, 2000.
- [101] A/Wiesler, H.Schober, R.Machauer and F.K.Jondral, "Software radio structure for UMTS and second generation mobile communication systems", IEEE 50th Vehicular Technology Conference, 1999. Pp.939-942.
- [102] J.P.Delahaye, J.Palicot and P.Leray, "A hierarchical modeling approach in software defined radio system design", IEEE Workshop on Signal Processing Systems Design and Implementation. Pp. 42-47, November, 2005.
- [103] M.L.Dickens, B.P.Dunn, and J.N.Laneman, "Design and Implementation of a Portable Software Radio", IEEE Communications Magazine, August, 2008.
- [104] T.Kohonen,"The Self-Organizing Map", Proceedings of the IEEE, Vol.78 No.9, September 1990.
- [105] Haykin, Simon (1998). Neural Networks: A Comprehensive Foundation (2ed). Prentice hall. ISBN 0132733501.
- [106] IEEE 802.22 Working Group, <http://grouper.ieee.org/groups/802/22/>
- [107] R.Natowicz, "Kohonen's Self-Organizing Maps for Contour Segmentation of Gray Level and Colour Images", Proceedings of the International Workshop on Artificial Neural Networks Computation. Pages: 890-897. Year of publication: 1995. ISBN:3-540-59497-3.
- [108] K.Tsagkaris, A.Katidiotis and P.Demestichas, "Enhancing Learning in Cognitive Radio Systems by means of Artificial Neural Networks", Poster Paper: Proceedings of ICST-Mobile Summit Conference 2009.
- [109] T.R. Newman, B.A.Barker, A.M.Wyglinski, A.Agah, J.B.Evans and G.J.Minden,"Cognitive engine implementation for wireless multicarrier transceivers", Wiley Wireless Communications and Mobile Computing, September, 2006.
- [110] M.Van, der Schaar and S.Shankar, "Cross-layer wireless multimedia transmission: challenges, principles and new paradigms", IEEE Wireless Communications, August 2005.
- [111] T. W. Rondeau et al., "Cognitive radio formulation and implementation," in Proc. of CROWNCOM, June 2006
- [112] T.R. et al., "Cognitive engine implementation for wireless multicarrier transceivers," Wiley Wireless Communications and Mobile Computing, Sept. 2006.

- [113] N. Baldo and M. Zorzi, "Fuzzy Logic for Cross-layer Optimization in Cognitive Radio Networks," IEEE Communications Magazine, vol. 46 T.W.
- [114] F.Ge, Q.Chen, Y.Wang and C.W.Bostain, "Cognitive Radio: From Spectrum Sharing to Adaptive Learning and Reconfiguration," IEEE Aerospace Conference, 2008.
- [115] I.A.Akbar, W.H.Tranter, "Dynamic Spectrum Allocation in Cognitive Radio using Hidden Markov Models: Poisson Distribution Case", IEEE 1-4244-1029-0/07 2007
- [116] J.C.Dunat, "Allocation Opportuniste de spectre pour les radios cognitives", PhD. Thesis l'ENST Paris, France. April, 2006.
- [117] Z.Zhang, "Intelligent Cognitive Radio : Research on Learning and Evaluation of CR based on Neural Network", IEEE CCC Code:1-4244-1430-X/07 2007
- [118] A.F. Cattoni, M.Ottonello, M.Raffetto and C.S.Regazzoni, "Neural Networks Mode Classification based on Frequency Distribution Features", IEEE CROWNCOM, 2007. USA.
- [119] T.Clancy and A.Khawar,"Security Threats to Signal Classifiers using Self-Organizing Maps", 4th IEEE CROWNCOM.
- [120] K.Tsagkaris, A.Katidiotis and P.Demestichas"Neural Network based learning schemes for cognitive radio systems" Computer Communications Journal Vol.31, issue 14, September, 2008. Pages 3394-3404
- [121] N.Baldo and M.Zorzi,"Learning and Adaptation in Cognitive Radios using Neural Networks", IEEE Consumer Communications and Networking Conference 2008,Las Vegas, USA.
- [122] N.Baldo,B.R.Tamma, B.S.Manoj, R.Rao and M.Zorzi,"A Neural Network based Cognitive Controller for Dynamic Channel Selection", IEEE, ICC-2009 Proceedings.
- [123] B.Ustundab and O. Orcay, "Pattern Based Encoding for Cognitive Communication", IEEE CROWNCOM 2008, Singapore
- [124] O.Orcay and B.Ustundab,"Pattern Recognition in Cognitive Communication", 23rd International Symposium on Computer and Information Sciences 2008.Turkey.
- [125] L.Bixio, M.Ottonello, H.Sallam, M.Raffetto and C.Regazzoni, "Signal Classification based on Spectral Redundancy and Neural Network Ensembles", IEEE CROWNCOM 2009.
- [126] R.Hachemani,J.Palicot and C.Moy"A New Standard Recognition Sensor for Cognitive Radio terminals", 15th European Signal Processing Conference (EUSIPCO 2007), Poland, September, 3-7, 2007
- [127] C.Roland and J.Palicot,"A New Concept for Wireless Reconfigurable Receiver", IEEE Com Mag.July, 2003.
- [128] <http://www.cis.hut.fi/projects/somtoolbox/>
- [129] M.Gandzitto,A.F.Cattoni,M.Musso and C.S.Regazzoni,"Distributed Cooperative Mode Identification For Cognitive Radio Mode Applications" XXVIIIth General Assembly of International URSI October, 23-29, 2005, New-Delhi, India.
- [130] M.E. Sahin and H.Arslan,"System Design for Cognitive Radio Communications", 1st International Conference on Cognitive Radio Oriented Wireless Networks and Communications, 2006. Location: Mykonos Island
- [131] J.M.Kim, S.H.Sohn, N.Han, G.Zheng, Y.Min Kim, J.K.Lee, "Cognitive Radio Software Testbed using Dual Optimization in Genetic Algorithm", 3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications, 2008.
- [132] N.Muhammad, K.Khalfallah, R.Knopp and R.Pacalet, "Reconfigurable DSP Architectures for SDR Applications", The 14th IEEE International Conference on Electronics, Circuits and Systems, 2007

- [133] M.Ali, Cavuslu, C.Karakuzu, S.Sahin, "Neural Network Hardware Implementation using FPGA", Page 287 – 290 3. ISEECE-2006
- [134] C.T.Huitzil, B.Girau and A.Gauffriau, " Hardware/Software Codesign for Embedded Implementation of neural Networks",
- [135] A.D. Rast, MM Khan, X.Jin, L.A.Plana and S.B. Furber, "A Universal Abstract-Time Platform for Real-Time Neural Networks", Proceedings of International Joint Conference on Neural Networks, Atlanta, Georgia USA, June, 14-19, 2009.
- [136] V.Valenta,R.Marsalek,G.Baudoin,M.Villegas,M.Suarez,F.Robert, "Survey on Spectrum Utilization in Europe: Measurements, Analyses and Observations", IEEE CROWNCOM 2010, France
- [137] Md Habibul Islam et al, "Spectrum Survey in Singapore: Occupancy, Measurements and Analyses", In Proceedings of CROWNCOM , 2008.
- [138] H. Lutkepohl, "New Introduction to Multiple Time Series Analysis", ISBN: 3-540-40172-5 Springer-Verlag, 2005.
- [139] H.Arslan, "Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems", Published by:Springer, 2007: ISBN:978-1-4020-5541-6.
- [140] Elman JL. "Finding Structure in Time." Cognit sci 1990; 14:179-211
- [141] T.Kolarik and G.Rudorfer, "Time Series Forecasting Using Neural Networks", Proceedings of the International Conference on APL: The language and its applications. Pages 86-94, Belgium. 1994.
- [142] M.Han, J.Xi, S.Xu, F.Yin, "Prediction of Chaotic Time Series Based on the Recurrent Predictor Neural Network", IEEE Transactions on Signal Processing, Vol.52, no.12, pp.3409-3416, 2004.
- [143] Q.Zhao and B.M.Sadler, "A survey of dynamic spectrum access", IEEE Signal Processing Magazine", vol.24, No.3, pp.79-89, May, 2007.
- [144] Q.Zhao, "Spectrum opportunity and interference constraint in opportunistic spectrum access", in Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), vol.3, April 15-20, 2007. Pp.605-608.
- [145] Q.Zhao, W.Ren and A.Swami, "Spectrum opportunity detection: How good is listen-before-talk?", in Proc. of the 41st Asilomar Conference on Signals, Systems and Computers, Nov. 4-7, 2007. Pp. 767-771.
- [146] H.S.Chen, W.Gao, and D.G.Daut, "Signature based spectrum sensing algorithms for IEEE 802.22 WRAN", IEEE International Conference on Communications, pp;6487-6492, 2007.
- [147] H.Urkowitz, "Energy detection of unknown deterministic signals", IEEE Vol.. 55, no.4, pages 523-531, 1967.
- [148] Yue Wen-jing, Zheng Bao-yu and Meng Qing-min, "Cyclostationary Property Based Spectrum Sensing Algorithms for Primary Detection in Cognitive Radio Systems", J.Shanghai Jiaotong Univ. (Sci), 2009. 14(6):676-680.
- [149] S.M.Mishra, R.W. Brodersen, S.T. Brink and R.Mahadevappa,"Detect and avoid: an ultra-wideband/WiMAX coexistence mechanism", IEEE Communications Magazine, vol.45. Topics in radiocommunications. No.6, pp.68-75. June, 2007
- [150] P.Papadimitratos, S.Sankaranarayanan and A.Mishra, "A bandwidth sharing approach to improve licensed spectrum utilization", IEEE Commum. Magazine, vol.43. no.12, pp.10-14, dec, 2005.
- [151] Z.Tian and G.B. Giannakis, "A wavelet approach to wideband spectrum sensing for cognitive radios", CROWNCOM 2007, Greece.

- [152] S.M.Kay,"Fundamentals of Statistical Signal Processing: Detection Theory, vol.2, Prentice Hall, Upper Saddle River, NJ,USA, 1998.
- [153] Y.Lin and C.He, "Subsection-Average Cyclostationary Feature Detection in Cognitive Radio", IEEE Int.Conference on Neural Networks and Signal Processing. June 8-10, 2008. China.
- [154] J.G.Proakis,"Digital Signal Processing:Prnciples, Algorithms and Applications", Fourth Edition. ISBN: 978-0131873742.
- [155] R.Chen and R.S.Tsay,"Nonlinear Additive ARX Models", Journal of the American Statistical Association, Vol.88, No.423.pp-955-967. September, 1993.
- [156] Tran N, Reed A, "ARIMA time series modeling and forecasting for adaptive I/O prefetching. Proceedings of the International Conference on supercomputing, pp.473-485. 2001.
- [157] Deshpande M, Karipys G, "Selective Markov Models for predicting web pages access", ACM Trans Internet Technol. 163-184. 2004.
- [158] M.Basso, L.Giarre, S.Groppi and G.Zappa," NARX Models of an Industrial Power Plant Gas Turbine", IEEE Transactions on control systems technology , vol.13, no.4, July, 2005.
- [159] T.Lin, C.L.Giles, B.Horne, S.Y.Kung, "A Delay Damage Model Selection Algorithm for NARX Neural Networks", IEEE Transaction on signal processing , Vol.45. No.11, November, 1997 Pp.2719-2730.
- [160] A.G. Parlos, O.T.Rais, A.F.Atiya, "Multistep ahead prediction using dynamic recurrent neural networks", Neural Networks 13(2000). 765-786.
- [161] Owens, A.J., "Empirical Modeling of Very Large Data Sets Using Neural Network", the IEEE-INNS-ENNS International Joint Conference on Neural Networks, July, 24-27, 2000. Volume 6. Page(s). 302-307.
- [162] R.Bakker, J.C. Schouten, C.Van den Bleek, C.L.Giles, "Neural Learning of Chaotic Dynamics: The error propagation algorithm", IEEE world conference on computational intelligence. p.2483, 1998.
- [163] Y.Chen, B.Yang, J.Dong, A.Abraham, "Time series forecasting using flexible neural tree model", information sciences 174, 219-235, 2005.
- [164] T.G.Barbounis, J.B.Teocharis, "Locally recurrent neural networks for long term speed and power prediction", Neurocomputing 69 2006, 466-496.
- [165] Cho K, Wang B, "Radial basis function based adaptive fuzzy systems and their application to system identification and prediction. Fuzzy sets Syst 83: 325-339. 1996.
- [166] Leung H, Lo T, wang S, "Prediction of noisy chaotic time series using an optimal radial basis function neural network", IEEE Trans. Neural Network. 12(5):1163-1172. 2001.
- [167] J.C. Principe, L. Wang, MA Motter (1998), "Local Dynamic Modeling with self-organizing maps and applications to nonlinear system identification and control. Proc IEEE 86(11).
- [168] D.R.Cox, "The regression analysis of binary sequences"Journal of Royal Statistical Society. Series B. vol.20, no.2, pp. 215-242, 1958.
- [169] A.Ranganathan, "The Levenberg-Marquardt Algorithm" Tutorial on LM Algorithm, June, 2004.
- [170] H.Demuth and M.Beale, "Neural Network Toolbox" for use with MATLAB. User's guide.
- [171] T.C.Clancy and B.D.Walker, "Predictive Dynamic Spectrum Access", SDR Forum Conference, 2006.
- [172] Z.Wang and S.Salous, "Time Series ARIMA Model of Spectrum Occupancy for Cognitive Radio", IET Seminar on Cognitive Radio and Software Defined Radios. September 2008, London, UK.

- [173] Z.Wang and S.Salous, "Spectrum Occupancy Statistics and Time Series Models for Cognitive Radio", Journal of Signal Processing Systems. Vol.62, Number 2, pp. 145-155.
- [174] M.Casdagli, "Nonlinear Prediction of Chaotic Time Series", Physica D: Nonlinear Phenomena, pp.335-356. Elsevier – 1989.
- [175] Z.Chen and R.C.Qiu, "Prediction of Channel State for Cognitive Radio Using Higher-Order Hidden Markov Model", Proceedings of the IEEE SoutheastCon March, 2010
- [176] C.H.Park, S.W.Kim, S.M.Lim and M.S.Song, "HMM based channel status predictor for cognitive radio", Proceedings of Asia-Pacific Microwave Conference. pp.1-4, December, 2007.
- [177] S.Yarkan and H.Arslan, "Binary Time Series Approach to Spectrum Prediction for Cognitive Radio", Proceedings of IEEE Conference on Vehicular Technology (VTC), pp.1563-1567, September, 2007.
- [178] V.K.Tumuluru, P.Wang and D.Niyato, "A Neural Network based Spectrum Prediction Scheme for Cognitive Radio", IEEE International Conference on Communications (ICC), South Africa, May. 2010.
- [179] H.Harada, "A Software Defined Cognitive Radio Prototype", IEEE 18th International Symposium on personal, Indoor and Mobile Radio Communications, 2007.
- [180] J.Mitola, "The Software radio Architecture", IEEE Communication Magazine, May, 1995.
- [181] Cognitive Radio Technology. Edited by Bruce Fette ISBN: L3: 978-0-7506-7952-7
- [182] P.Pawalczak, "Technical Challenges of Cognitive Radio-Related Systems" CRNI Conference :Radio Spectrum Management. 2008.
- [183] **Best Paper Award** M.I.Taj, O.Hammami and K. Huggins, "Performance Evaluation of SDR on embedded platform: The case of OSSIE" The 2nd IEEE International Conference on Computer, Communications and Control, 2009. Pakistan.
- [184] M. I.Taj, K. Huggins and O.Hammami, "OSSIE Signal Processing Functions Performance Enhancements through Parallelization in an embedded multiprocessor architecture" 2009 Software Defined Radio Technical Conference and Product Exposition, Madrid, Spain, April,2009.
- [185] M.I.Taj, O.Hammami and M.Akil,"SDR waveform components implementation on single FPGA Multiprocessor Platform" The 17th IEEE International Conference on Electronics, Circuits, and Systems, 2010. Greece.
- [186] M.I.Taj, M.Akil and O.Hammami "Standard Recognizing Self Organizing Map based Cognitive Radio Transceiver", The 5th IEEE International Conference on Cognitive Radio Oriented Wireless Networks and Communications, 2010, Cannes, France.
- [187] M.I.Taj, and M.Akil, "Standard Recognizing Artificial Neural Networks Based Reconfigurable Cognitive Radio Transceiver and its Embedded Implementation." IET Journal: Special issue on Cognitive Communications. *Under Review*
- [188] M.I.Taj and M.Akil, "Cognitive Radio Spectrum Evolution Prediction using Artificial Neural Networks based Multivariate Time Series Modelling." The 17th IEEE European Wireless Conference, April, 2011. Vienna, Austria.
- [189] M.I.Taj and M.Akil, "Cognitive Radio Multiple-step ahead Spectrum Evolution Prediction using Artificial Neural Networks based Multivariate Time Series Modelling." Journal of Signal and Information Processing. *Submitted.*