ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

ADAPTIVE SYMBOL GLOSSARY FOR PATTERN BASED COGNITIVE COMMUNICATION SYSTEM

Ph.D. THESIS

Husam Y. I ALZAQ

Department of Computer Engineering

Computer Engineering Programme

NOVEMBER 2019

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

ADAPTIVE SYMBOL GLOSSARY FOR PATTERN BASED COGNITIVE COMMUNICATION SYSTEM

Ph.D. THESIS

Husam Y. I ALZAQ
(504112512)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Assoc. Prof. Dr. Burak Berk ÜSTÜNDAĞ

NOVEMBER 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

ÖRÜNTÜ TABANLI BİLİŞSEL HABERLEŞME SİSTEMİ İÇİN UYARLAMALI SEMBOL SÖZLÜĞÜ

DOKTORA TEZİ

Husam Y. I ALZAQ
(504112512)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Assoc. Prof. Dr. Burak Berk ÜSTÜNDAĞ

KASIM 2019

Husam Y. I ALZAQ, a Ph.D. student of ITU Graduate School of Science Engineering and Technology 504112512 successfully defended the thesis entitled "ADAPTIVE SYMBOL GLOSSARY FOR PATTERN BASED COGNITIVE COMMUNICATION SYSTEM", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Assoc. Prof. Dr. Burak Berk ÜSTÜNDAĞ** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. Fatih ALAGÖZ** Boğaziçi University	.....
	Assoc. Prof. Dr. Berk CANBERK Istanbul Technical University	.....
	Assist. Prof. Dr. Yusuf YASLAN Istanbul Technical University	.....
	Assoc. Prof. Dr. Hacı İLHAN Yildiz Technical University	.....

Date of Submission : **5 July 2019**
Date of Defense : **14 November 2019**

To my parents,
To my spouse and children,
To the martyrs of honor, dignity, and democracy,

FOREWORD

No work is complete without a word of appreciation and gratitude for the guidance, sincere wishes, assistance and support that created a firm foundation.

First and foremost, I would like to express my deepest gratitude to my supervisor Assoc. Prof. Dr. B. Berk ÜSTÜNDAĞ not only for his guidance, criticism, encouragements and insight throughout this research but also for his continuous confidence in me, and his unforgettable and valuable contributions to my career. His continuous motivation helped me sail through this voyage and reach the final destination.

My faithful wife, Tahani, for holding fort while I was away from home in pursuit of this degree, and my children, Yunus, Büşra, Saed and Arwa, for living their babyhood in absence of their father. You all deserve an honorary degree.

I find no words to acknowledge the sacrifices, love and affection of my wonderful parents, for their patience, encouragement and continuous morale support. They supported me in every possible way and in every single moment during these years. In spite of the closing boarders that blocked us to meet, they were always beside me. They gave me the strength and courage to continue my effort in hard times.

NOVEMBER 2019 Husam Y. I ALZAQ

TABLE OF CONTENTS

Page

ABBREVIATIONS

ADC : Analog-to-Digital Converter
AFPSK : Amplitude Frequency Phase Shift Keying
AI : Artificial Intelligent
AM : Amplitude Modulation
AMC : Automatic Modulation Classification
AMR : Automatic Modulation Recognition
ANN : Artificial Neural Network
ASK : Amplitude Shift Keying
ASIC : Application-Specific Integrated Circuit
AWGN : Additive white Gaussian noise
BER : Bit Error Rate
BN : Bayesian network
BP : Back Propagation Algorithm
BPSK : Binary Phase Shift Keying
BR : Bayesian regularization Back-propagation Algorithm
CE : Cognitive engine
CP : Cyclic Prefix
CR : Cognitive Radio
CRN : Cognitive Radio Network
CWT : Continuous Wavelet Transform
DAC : Digital-to-Analog Converter
dBc : Decibels relative to the Carrier
dBm : Decibels relative to 1 mW
DDC : Digital Down Converter
DFT : Discrete Fourier Transform
DSB : Double Sideband Modulation
DSP : Digital Signal Processing
DSSS : Direct Sequence Spread Spectrum
DUC : Digital Up Converter
DWT : Discrete Wavelet Transform
FANN : Feedforward Artificial Neural Network
FDM : Frequency Division Multiplexing
FFNN : Feedforward Artificial Neural Network (similar to FANN)
FFT : Fast Fourier Transform
FHSS : Frequency Hopping Spread Spectrum
FFT : Fast Fourier Transform
FM : Frequency Modulation
FMC : FPGA Mezzanine Card
FSK : Frequency Shift Keying
FPGA : Field-Programmable Gate Array

GCD : Greatest Common Divisor
GDX : Gradient Descent with Momentum and Adaptive Learning Rate Algorithm
GS : Glossary selector
HDL : Hardware Description Language
HPC : High Pin Count
HPF : High Pass Filter
ICI : Inter-Channel Interference
IFFT : Inverse Fast Fourier Transform
ISI : Inter-Symbol Interference
ISM : Industrial Scientific and Medical
LPF : Low-Pass Filter
LM : Levenberg-Marquardt Back-propagation Algorithm
LSB : Lower Sideband Modulation
LSE : Link Spectral Efficiency
***k*-ASK** : $k-$bit Amplitude Shift Keying
***k*-FSK** : $k-$bit Frequency Shift Keying
***k*-PSK** : $k-$bit Phase Shift Keying
***k*-QAM** : $k-$bit Quadrature amplitude Modulation
MAC : Media Access Control
MIMO : Multiple Input Multiple Output
ML : Machine Learning
MLP : Multilayer Perceptron Neural Network
MSE : Mean Squared Error
MSK : Minimum Shift Keying
OFDM : Orthogonal Frequency-Division Multiplexing
OOK : On-off Keying
PBCCS : Pattern Based Cognitive Communication system
PU : Primary User
QAM : Quadrature Amplitude Modulation
QPSK : Quadrature Phase Shift Keying.
QoS : Quality of Service
SCG : Scaled Conjugate Gradient Back-propagation Algorithm
SDR : Software Defined Radio
SNR : Signal to Noise Ratio
SPEC : Sinusoidal Pattern Envelop Construction
SU : Secondary User
SVM : Support Vector Machine
TDANN : Time Delay ANN
UHF : Ultra High Frequency
USB : Upper Sideband Modulation
USRP : Universal Software Radio Peripheral
VSB : Vestigial Sideband
WNN : Wavelet Neural Network

SYMBOLS

t	**:** Time
$a \times b$	**:** The memory word-size
l	**:** Number of DWT levels
M	**:** Maximum range of P_n
N	**:** Number of filter tap
h_k	**:** The low-pass k^{th} filter coefficient
m	**:** Number of magnitude bits
m_i	**:** The i^{th} moduli of P_n
n	**:** The moduli-set base (e.g. P_7)
q	**:** Number of RNS channel
τ	**:** Latency
w	**:** Memory word-length
y	**:** Input scaling factor
z	**:** Filter scaling factor

LIST OF TABLES

Page

LIST OF FIGURES

Page

ADAPTIVE SYMBOL GLOSSARY
FOR PATTERN BASED COGNITIVE COMMUNICATION SYSTEM

SUMMARY

Wireless communications systems are experiencing an exponential growth and becoming an essential element of our daily life. The advancement in wireless communication standards focus on increasing the datarate transmission, improving reliability, and reducing the latency, while the spectrum efficiency and the robustness to low-SNR have so far received little attention. Pattern-based cognitive communication system (PBCCS) was designed to allow the spectrum to be used more efficiently by adapting the transmitter behavior to the environmental conditions and the users requirements. Indeed, PBCCS is a leap approach from using model-based algorithms to data-driven model, which merges the methods of computational intelligence and machine learning with software defined radio (SDR), which has opened the road to realizing signal processing and communication algorithms using software components.

PBCCS is a cross-layer approach that merges the channel encoding and adaptive modulation techniques. The main feature of PBCCS is the construction of optimal communication signals that can be recovered by the receiver at low-SNR. The optimal communication signals, known as symbols, encode data by mapping the bit string to a patterned symbol. Symbols of similar band characteristics form a glossary. The glossary dataset contains various glossaries that are adapted to the communication requirements such as high bandwidth, low-SNR or high spectral efficiency. The transmitter selects the appropriate glossary from the glossary dataset that maximizes the link spectral efficiency (LSE) value with respect to the lowest SNR level. As a result, the transmitter is not only responsible for managing channels but also selecting an appropriate pattern from the glossary. Therefore, the cognitive coding algorithm that is employed at the transmitter, is capable of adapting its behavior accordingly with regards to the learning parameter of the environment. At severe conditions, the cognitive radio's transmitter switches to a new optimal cognitive code.

From another perspective, the receiver that had prior knowledge on the transmitted information can correctly recover the data. The receiver pre-processes the received signal by extracting a limited set of wavelet features. Then, the extracted features are fed into an artificial neural network (ANN) to recover the digital data carried by the distorted symbol. The aim of using ANN at the receiver is to predict the original bits of the distorted received signal. The receiver does not construct the similar analog signal or estimate its parameter. Instead, it classifies the extracted samples to a known pattern, so that the correct bits can be inferred.

The first part of this dissertation focused on the design and implementation of PBCCS. The algorithms that are used in the implementation and related configuration are built up in MATLAB environment and the performance of system was observed with a simulator developed for this purpose. The experimental approach helps us

to investigate the performance of PBCCS by employing different wavelet families and several ANN configuration. Furthermore, the transmitter and receiver FPGA are implemented using Xilinx System Generator Toolbox for MATLAB Simulink environment. An experimental set-up to achieve reliable output was prepared using Xilinx Zynq-7000 (XC7Z045) board with an integrated transceiver chip AD9361.

Results have highlighted some aspects of PBCCS implementations. To summarize the major key output of this study: (1) The proposed model increases the spectral efficiency by constructing a secure and non-periodic communication signals. (2) the optimized signal patterns that are decoded solely by DWT preprocessed signals and artificial neural network have been decoded at SNR of -5 dB of a bit error rate (BER) of 10^{-5} under an Additive White Gaussian Noise (AWGN) channel.

In the second part of this dissertation, the design of DWT on FPGA was proposed. Rather than implementing DWT via multipliers and an adder tree, a multiplier-free architecture was suggested due to their resultant in low complexity systems and their high throughput processing capability. Two famous multiplier-free architecture were presented — i.e., the Distributed Arithmetic Algorithm (DAA) and Residue Number System (RNS). We have studied the effect of multiplier-less architectures DWT, which has a substantial influence on the overall performance of the design and resource availability. Moreover, we addressed the effect of increasing the number of filter taps and word-length, which have a substantial influence on the overall performance of the design and resource availability. The main finding was that the DAA-based approach is appropriate for a small number of filter taps, while the RNS-based approach would be more appropriate for more than 10 filter taps, yet both DAA- and RNS-based offer high signal quality with peak signal-to-noise ratio (PSNR) of 73.5 and 56.5 dB, respectively.

The third part is an extension of the second part, where an optimized multi-level DWT was proposed. It presents the implementation of RNS-based optimized multiplier-less two-level DWT. It also presented an analytical comparison among different DWT implementations as well as the performance results. This approach intensively use memory to speed up the entire processing time. In order to achieve low latency, we incorporated two novel ideas into the two-level proposed design, as follows: (1) eliminating the intermediate Residue-to-Binary Converter (RBC) unit; (2) replacing the internal memory of the second level by simple circular shift operations. The overall results of the experiments showed that there is a considerable improvement when the proposed two-level DWT design is used with regard to latency and peak signal-to-noise ratio (PSNR) precision value at the final output.

ÖRÜNTÜ TABANLI BİLİŞSEL HABERLEŞME SİSTEMİ İÇİN UYARLAMALI SEMBOL SÖZLÜĞÜ

ÖZET

Kablosuz iletişim sistemleri, üstel bir büyüme yaşamakta ve günlük hayatımızın önemli bir unsuru haline gelmektedir. 4G ve 5G mobil ağlar gibi kablosuz iletişim standartlarındaki ilerlemeler, veri iletim hızını arttırmaya, güvenilirliği geliştirmeye ve gecikmeyi azaltmaya odaklanırken, spektrum verimliliği ve düşük-SNR'ye karşı dayanıklılık yeterince ilgi görmemiş ve etkisi iyi araştırılmamıştır. Gürültüye karşı dayanıklılığı arttırırken, spektral bant genişliğini kullanmanın kritik olduğu açıktır.

Son yıllarda, kablosuz iletişim sistemlerinin otomatik olarak öğrenilen özellikler ile genişletilmesiyle birlikte spektral bant genişliğinin kullanımı konusunda çok daha yüksek performans elde edilmiştir. Kablosuz iletişim alanında geniş çaplı sorunlarda yapay sinir ağı (YSA - Artificial Neural Network, ANN) ve makine öğrenme teknikleri ele alınmaktadır. Bunları göz önünde bulundurarak, bu tez çalışmasında düşük SNR'nin etkisini azaltan ve gürültüye karşı dayanıklılığı arttıran ANN çözümünü uyarladık.

Shannon Limit Teoremi (Shannon, 1948), kanal veri iletim kapasitesinin, C (bps), kanalın İşaret Gürültü Oranı (SNR) ve kullanılan spektral bant genişliğine, BW (Hz), direk bağlı olduğunu gösterir. SNR, bağlantı spektral verimliliğini (LSE) $\eta = C/B$ (bps/Hz) etkileyen en önemli faktördür ve kanal kapasitesini sınırlar. Bu nedenle, kanal kapasitesini en üst düzeye çıkarmak için, spektral bant genişliğinin ve düşük SNR seviyesine olan dayanıklılığın kullanılması önemlidir. Mevcut modülasyon teknikleri, yaklaşık sıfır BER ile spesifik SNR seviyeleri (0 dB'den daha büyük) altında veri iletim kapasitesini (LSE) maksimize eder. Ancak, hala Shannon Limit'ten uzaktırlar ve düşük SNR'nin etkisini göz ardı etmektedirler. Bu nedenle, kanal kapsitesinin maksimizasyonunda, spektral band genişliğinin verimli kullanımı ve yüksek gürültü değerlerine dayanıklılık en önemli etkenlerdir.

Önerilen ve uygulama başarımı verilen karıştırıcıya dayanıklı haberleşme sisteminde bilişsellik iki yönüyle etkindir. Birincisi, radyo frekanslı (RF) haberleşme bandındaki gürültü seviyesine göre doğrudan kodlama ve sayısal işaret modülasyonunun veri karşılığını temsil eden sembollerin örüntü sözlüğünün anında kullanılmasıdır. İkincisi ise haberleşme spektrumunun yoğunluk ve gürültü durumunun algılanması ve buna göre dayanıklılık ihtiyacı ile iletişim bant genişliği arasında optimizasyonu sağlayan örüntü sözlüğünün seçilmesidir. Bilişsel kodlama, yazılım tabanlı radyo uygulamalarında yaygın olarak bilinen bilişsel iletişimden farklı olarak spektrum verimliliğini arttırmakla kalmayıp, değişen örüntü sözlükleri ile karıştırıcı gücüne göre dayanıklılığı da arttırmaktadır. Karıştırıcı etkisi azaltılarak erişilen artırılmış SNR seviyesi, iletişimin sürdürülebilirliği bakımından kritik öneme sahiptir.

PBCCS, kanal kodlaması ve uyarlamalı modülasyon tekniklerini birleştiren, katmanlar arası bir yaklaşımdır. PBCCS'nın vericisi, bit dizisini bir sinyal sembolüne eşleyerek

kodlar ve semboller olarak bilinen optimal iletişim sinyallerini oluşturur. Sözlük veri kümesi, yüksek bant genişliği, düşük SNR veya yüksek spektral verimlilik gibi iletişim gereksinimlerine uyarlanmış farklı özelliklere sahip çeşitli sözlükler içerir. Bağlantı Spektral Verimliliğini (LSE) maksimize edecek şekilde uygun sözlük, en düşük SNR seviyesine göre sözlük veri kümesinden seçilir. Sonuç olarak, gönderici (verici) sadece kanalların yönetilmesinden sorumlu olmayıp, aynı zamanda sözlükten uygun bir sembol de seçmektedir. Bu nedenle, vericide kullanılan bilişsel kodlama algoritmasının davranışı, çevrenin öğrenme parametresine göre uyarlanabilmektedir. Şiddetli ve ağır koşullarda, bilişsel radyo vericisi yeni bir optimal bilişsel kodu seçecektır. Diğer açıdan, iletilen veriler hakkında önceden bilgisi olan alıcı, verileri doğru biçimde kurtarabilir. Çok seviyeli ayrık wavelet dönüşümü (Discrete Wavelet Transforem, DWT) kullanarak alıcı, sınırlı bir wavelet özellik seti çıkararak alınan sinyali önceden işlemden geçirir. DWT'den elde edilen veriler, bozuk sembol tarafından taşınan dijital verileri kurtarmak için yapay sinir ağı tarafından kullanılmaktadır. DWT ile ANN kullanmanın amacı, alıcı tarafından alınan bozulmuş sinyalin orijinal bitlerini tahmin etmektir. Alıcı, benzer analog sinyali oluşturmaz veya parametresini tahmin etmez. Bunun yerine, çıkarılan örnekleri bilinen bir sembole göre sınıflandırır, böylece doğru bitler çıkarılabilir.

Uygulamada ve ilgili konfigürasyonda kullanılan algoritmalar, ilk olarak MATLAB ortamında oluşturulmuş ve geliştirilmiş bir simülatör ile test edilmiş ve sistemin performansı gözlenmiştir. Gönderci (Verici) ve alıcı FPGA uygulamalarının geliştirilmesinde MATLAB Simulink ortamının bir aracı olan Xilinx System Generator Toolbox kullanılmıştır.

Tez katkısı. Bu tez çalışmasının ilk araştırma katkısı, ilave bir beyaz Gauss gürültüsü (AWGN) kanalında PBCCS'nin performansını deneysel olarak araştırmayı içermektedir. PBCCS, alınan sinyali ve sinir ağını işlemek için çok seviyeli DWT kullanır. Birkaç sözlük oluşturan bir algoritma geliştirlmiştir. Ayrıca, bilişsel alıcı için uygun iletişim modellerini ve alınan semboller üzerindeki etki faktörlerini inceledik. Ek olarak, YSA'nın boyutunu azaltmak için 4 ve 5 seviyeli DWT kullanmanın etkisini inceledik. Son olarak, azaltılmış bir YSA yapısı sergileyen alıcının uzay karmaşıklığı analiz edildi.

İkinci araştırma katkımız, alınan sinyali önceden işlemek için kullanılan çok düzeyli DWT uygulamasını incelemektir. Simülasyonumuz sayesinde çok seviyeli DWT'nin son tasarım üzerindeki etkisini fark ettik. DWT ve yapay sinir ağının FPGA içindeki çarpan kaynaklarının kullanımı için rekabet halindedirler. Bu nedenle, yapay sinir ağının FPGA'daki mevcut çarpan kaynakları kullanmasına ve DWT'nin de kalan sayı sistemi (RNS) ve dağıtılmış aritmetik algoritma (DAA) gibi bellek-tabanlı teknikleri kullanmasına karar verdik. Aslında, RNS kullanan DWT uygulaması yeni bir konu değildir, ancak çalışmamız çok seviyeli DWT'nin net bir uygulamasını ilk sunan çalışmadır. Özellikle, birinci seviyede DWT'yi etkili bir şekilde kullanan ve sonraki seviyelerde herhangi bir bellek elemanı kullanmayan iki seviyeli bir RNS tabanlı DWT tasarımı önerdik. Ayrıca bu tasarım, ardışık seviyeler arasında çoklu kalıntılardan ikili transformatörlerin (RBC) kullanımını ortadan kaldırmaktadır. Genelde, seviyelerin sayısı çıkış kelimesi uzunluğu ile sınırlıdır ve matematiksel olarak belirlenir. Son olarak, daha az güç tükecek şekilde önerilen RNS tabanlı basit donanım yapısıyla, yüksek PSNR değerlerine ulaşılmaktadır.

Tez organizasyonu. Bu tezin ilk kısmı, PBCCS'nin tasarım ve uygulamasına odaklanmıştır. Uygulamada ve ilgili konfigürasyonda kullanılan algoritmalar MATLAB ortamında oluşturulmuş ve bu amaçla geliştirilen bir simülatör ile sistemin performansı gözlenmiştir. Deneysel yaklaşım, farklı dalgacık aileleri ve çeşitli ANN konfigürasyonları kullanarak PBCCS'nin performansını araştırmamıza yardımcı olmaktadır. Ayrıca, verici ve alıcı FPGA, MATLAB Simulink ortamı için Xilinx Sistem Jeneratör Araç Kutusu kullanılarak uygulanmaktadır. Güvenilir bir çıktı elde etmek için deneysel bir kurulum, entegre bir alıcı verici çip AD9361 ve Xilinx Zynq-7000 (XC7Z045) kartı kullanılarak hazırlandı.

Tezin ikinci bölümünde, FPGA üzerinde DWT tasarımı önerilmiştir. DWT'yi çarpanlar ve bir toplayıcı ağacı ile uygulamak yerine, çarpan içermeyen bir mimari düşünüldü ki düşük karmaşıklık sistemlerinde elde edilen sonuçlara ve yüksek verimli işleme kapasitesine neden olabilsin. Bu bağlamda, iki ünlü çarpan içermeyen mimari sunuldu ve onlar Dağıtılmış Aritmetik Algoritma (DAA) ve Kalıntı Sayı Sistemi (RNS). Tasarımın ve kaynak kullanılabilirliğinin genel performansı üzerinde önemli bir etkiye sahip olan, çarpanı olmayan mimariler üzerinde DWT'nin etkisini inceledik. Dahası, tasarım ve kaynak kullanılabilirliğinin genel performansı üzerinde önemli bir etkiye sahip olan filtre katsayılarının ve kelime uzunluğu sayısının artırılmasının etkisini ele aldık. Bu araştırmanın ana sonucu, DAA-tabanlı yaklaşım az sayıda filtre katsayıları için uygun iken, RNS tabanlı yaklaşımın 10'dan fazla filtre katsayıları için daha uygun olduğu, ayrıca hem DAA hem de RNS-tabanlı yaklaşımın yüksek sinyal kalitesini sırasıyla 73.5 ve 56.5 dB pik sinyal-gürültü oranı (PSNR) ile sergilediğidir.

Tezin son bölümü, ikinci bölüm için bir genişletmedir ve burada optimize edilmiş çok seviyeli bir DWT önerilmiştir. Bu bölüm, RNS-tabanlı, optimize edilmiş, çoğaltıcı olmayan, iki seviyeli DWT'nin gerçeklemesini sunar. Ayrıca farklı DWT uygulamaları ve performans sonuçları arasında analitik bir karşılaştırma sunmaktadır. Bu yaklaşım tüm işlem süresini hızlandırmak için belleği yoğun bir şekilde kullanır. Düşük gecikme süresi elde etmek için, iki seviyeli tasarıma belirttiğimiz eklemeleri yaptık: (1) orta RBC ünitesinin ortadan kaldırılması ve (2) basit dairesel kaydırma işlemleri ile ikinci seviyenin dahili belleğinin değiştirilmesi. Ölçümlerimizin sonuçları, önerilen iki seviyeli DWT tasarımının, gecikme ve tepe sinyal-gürültü oranı (PSNR) değeri ile kullanıldığında kayda değer bir gelişme olduğunu göstermiştir.

1. INTRODUCTION

Wireless communications systems are experiencing an exponential growth and becoming an essential element of our daily life. The advancement in wireless communication standards, such as 4G and 5G mobile networks, focuses on increasing the data-rate transmission, improving reliability, and reducing the latency, while the spectrum efficiency and the effect of low-SNR is not well studied. Some of th bottlenecks for next generation networking wireless systems (e.g. 5G) includes the increasing number of devices with increased energy consumption, shortage of spectrum, and mobility issues. It is quite obvious that the utilization of spectral bandwidth is highly crucial while improving the robustness against surrounding noise.

With the advance of software-defined radio (SDR) technology, wireless communications systems have been extended by automatically learned features, yielding much higher performance on utilizing the spectral bandwidth. Artificial neural network (ANN) and machine learning techniques are addressed in wide range problem in wireless communication field. In this context, we adapted an ANN solution that mitigates the effect of low-SNR and improves the robustness against noise.

This dissertation design and develop the first prototype of pattern-based cognitive communication system (PBCCS). In fact, PBCCS is inspired by the recognition capability of humans to concentrate on a single conversation irrespective of the surrounding loudness. Once human ears hear sounds from different sources, the brain chooses to pay attention to a particular voice stream amongst a whole range of sound streams in the nearby environment.

In this study, we analyzed the architecture and performance of the prototype, both in a simulated environment and real test-bed. This architecture incorporates a wavelet-preprocessed artificial neural network (ANN). Additionally, we described and discussed the transceiver of PBCCS, when implemented on real FPGA. The cognitive transmitter is capable of adapting its behavior according to the learning parameter

of the environment. Invariant to a noisy band, the cognitive receiver that had prior knowledge on the transmitted information can correctly recover the data.

1.1 Motivation

The advancement in software-defined radio (SDR), has moved the signal processing at the transceiver of a wireless communication from hardware to software components. In particular, SDR comprises of a programmable communication system where functional behavior can be made by merely updating the software's parameters.

With the assistance of the software components and its capability to adapt its working point, the transceiver would overcome high error rate and low capacity in low-SNR values. Due to weather conditions or long-distance communication, the received signals often experience low-level SNR. The problem even get worsen for underwater acoustic communications, where the received signal undergoes harsh underwater environment. Moreover, it is necessary for some military applications to operate in low-SNR to avoid detection and eavesdropping. In the meantime, as SNR values decreases, bit error rate(BER) needs to be minimized. Therefore, the motivation of this thesis is practically to increase the spectral efficiency of communication systems under low-SNR level for better communications.

1.2 Purpose of Thesis

The objective of this dissertation is to explore the applicability of using pattern-based cognitive system in wireless communication. We have outlined the new heights of innovation in applying preprocessed signal patterns and neural network. In addition, this dissertation aims at closing the productivity gap between the simulation and design of the PBCCS on FPGA systems. In particular, while evaluating the robustness of the proposed system against low-level SNR in the communication channel, our ultimate goal of this dissertation focuses on transceiver HW design issues.

1.3 Contributions of this Research

As mentioned in the previous sections, we have developed a pattern-based cognitive system that is able to recognize the received data by using a pre-processed wavelet and ANN.

The first research contribution involves experimentally investigating the performance of PBCCS in an additive white Gaussian noise (AWGN) channel by employing multi-level DWT to preprocess the received signal and neural network. We developed an algorithm that constructs several glossaries. In addition, we studied the appropriate communication patterns for the cognitive receiver and the influence factors on the received symbols. We examined the effect of using 4- and 5-level DWT, which reduced the size of the ANN. Finally, the space complexity of the receiver that exhibits a reduced ANN structure was analyzed.

Our second research contribution examines the multi-level DWT implementation that is used to preprocess the received signal. From our simulation, we have noticed the influence of multilevel DWT on the final design. Owing to competition on the multiplier resources inside the FPGA between DWT and neural network, it is a high degree of importance to resolve this competitive conflict. Therefore, we have decided to implement the neural network with the available multiplier resources and also to implement the DWT by multiplier-less techniques such as residue number system (RNS) and distributed arithmetic algorithm (DAA). Though, the implementation of DWT using RNS is not a new topic, but our work was the first to provide an explicit implementation of multi-level DWT. In particular, we proposed a new design of two-level RNS-based DWT that efficiently uses the memory elements in the first-level DWT and which do not employ any memory element in the next levels. In addition, this design eliminates the use of multiple residue-to-binary converters (RBCs) between consecutive levels. Generally, the number of level is bound by the output word length and we have determined it mathematically (Eqs. 4.16 and 4.17). Finally, the proposed RNS-based approach could achieve high PSNR values with simple hardware structure that consumes less power.

1.4 Thesis Organization

This dissertation contains three published papers and the content of each is devoted as a standalone chapter.

Chapter 2 focuses on the design of pattern-based cognitive communication system. It describes the structure of the PBCCS model and its internal components in detail. It also presents the methodology that is employed in extracting the signal feature and artificial neural network that is being used as a classifier. The chapter highlights the training and testing of the classifier as well as the performances of the PBCCS.

Chapter 3 describes the implementation of discrete wavelet transform without employing multipliers. It reviews the theoretical background of two famous multiplier-less techniques— i.e., DAA and RNS. It starts by providing a general preliminary information on this techniques. We further showed an analytical comparison between these approaches as well as the performance results.

Chapter 4 extends the previous chapter and focuses on multi-level DWT optimization. It presents the implementation of the optimized multiplier-less two-level DWT that is based on RNS. It also presents an analytical comparison among different DWT implementations as well as the performance results.

Chapter 5 introduces the development of a PBCCS test-bed. Within this scope, signal detection scheme that incorporates learning techniques is presented in this chapter. After that, the evaluations of PBCCS are presented at the end of the chapter.

Finally, some concluding remarks are summarized in Chapter 6. It also wraps up the most important concepts, highlights major findings, along with the suggestion of future research directions and extensions.

2. VERY-LOW-SNR COGNITIVE RECEIVER BASED ON WAVELET PREPROCESSED SIGNAL PATTERNS AND NEURAL NETWORK[1]

2.1 Introduction

The efficiency of bandwidth utilization takes an important role in spectrum management [1, 2]. Due to fixed spectrum assignment policies and its inadequate to meet an unexpected increase in the number of higher-data-rate devices, the spectrum is inefficiently used. Cognitive radio (CR) [3–6] was proposed as a promising solution to alleviate the spectrum scarcity problem through dynamic management of the available spectrum. The pioneer work of Mitola *et al.* [3] led to an efficient utilization of the spectral bandwidth by allowing the secondary user (SU), who is not serviced, to detect and access the primary network spectrum gaps. CR allows detection of the state of the spectrum to adjust its own system parameters (transmission power, frequency band, throughput and modulation scheme) in real time [7]. The result is that the utilization of the spectral bandwidth is performed with the software flexibility in an adaptive manner with respect to the system parameters.

However, efficient spectral bandwidth usage under the influence of higher noise is not the major consideration of CR. Claude Shannon [8] showed that the SNR is a leading factor that influences the link spectral efficiency (LSE), $\eta = C/B$, (in bps/Hz). SNR also limits the channel capacity. Therefore, the utilization of spectral bandwidth and the robustness to high SNR level are the keys to maximize the channel capacity.

Thus, a pattern-based cognitive communication system (PBCCS) was introduced to optimize the overall spectral efficiency with respect to SNR [9, 10]. It is inspired by the recognition capability of humans to concentrate on a single conversation irrespective of the surrounding loudness. If human ears hear sounds from different sources, the brain chooses to pay attention to a particular voice amongst a whole range of sound streams

[1]This chapter is based on the paper "Alzaq H.Y., Ustundag B.B., Very-low-SNR cognitive receiver based on wavelet preprocessed signal patterns and neural network, EURASIP Journal on Wireless Communications and Networking, 2017(1): 120"

in an environment. Similar to human cognitive capabilities, the communication system in PBCCS selectively recognizes and recovers the communication signal(s) a into known symbol(s), even within the same frequency range.

2.1.1 Related work

Conventional cognitive radio is equipped with various techniques for making wireless systems more flexible and robust to channel variation. Mitola, in his dissertation [11], stated that, although many aspects of wireless networks are artificial, they may still be enhanced by machine learning (ML). Recently, machine learning algorithms have become one of the key enabling features of cognitive radio in many applications. In previous literature, many techniques and algorithms have been applied to the cognitive radio engine [12, 13], such as the artificial neural network (ANN), hidden Markov model (HMM), fuzzy logic control, meta-heuristic algorithms (evolutionary/genetic algorithm) and rule-based systems [14–16].

On the other hand, the PBCCS structure is an extension for CR, which is a cross-layer architecture. The control unit of PBCCS is located in the data-link layer and communicates with the external glossary space that manages the transmission process. Table 2.1 summarizes the similarities and differences between PBCCS and cognitive radio.

Generally, ANN has been adapted in the cognitive radio engine for various modulation classification, known as automatic modulation recognition (AMR) or automatic modulation classification (AMC). For instance, ANN was implemented for channel sensing [17–19] and spectrum prediction [17, 20], *etc*. To enhance ANN classification accuracy, ANN is usually combined with the extracted features from the received signal, which allows the engine to have the capability to identify the modulation scheme at low SNR levels. Cyclic spectral analysis [17], wavelet cyclic features [21], temporal feature-based modulation [22, 23], carrier frequency and baud rate [24], and continuous wavelet transform (CWT) [25] are some examples of these features. Dahap *et al.* [26] proposed a digital recognition algorithm that uses six features extracted from instantaneous information and signal spectrum to discriminate between different modulated signals. Table 2.2 shows a brief summary of these approaches. However, most of the aforementioned approaches have been used to

Table 2.1 : Comparison between Pattern Based Cognitive Communication System and Cognitive Radio.

	PBCCS	Cognitive Radio
Objective	To improve data transmission performance under bandwidth limitations by maximizing the LSE value with respect to SNR level.	To efficiently manage the spectrum by using the unoccupied spectrum band when it is not used by the licensed user.
Transmitter Side	It selects one signal pattern from the glossary space in adaptive manner.	It uses the frequency bandwidth in an adaptive manner with respect to channel availability.
Receiver Side	The distorted signal is recovered by a trained ANN. Recovering the information without separate demodulation, decoding and error recovery operations.	Receiver must have a wide-band front-end to detect spectrum holes. It should take into account the adaptive modulation feature, since some receivers employs AMC. Many ML algorithms can be used.
Modulation Technique	AFPSK	QAM, QPSK, BPSK, ...*etc*.
Cognition Manager	Glossary selector (GS)	Cognitive engine(CE)

classify low-order modulation technique, such as 2-ASK and 2-PSK. In addition, a prior signal information such as the carrier frequency is required. Additionally, if these approaches classify high-order modulation schemes, they construct a large ANN. The PBCCS is a kind of AMC that has not only the ability to classify high-order modulation, but also can encode the received analog signal at very low SNR.

In this work, we choose to use the ANN model at the PBCCS receiver owing to its powerful capabilities. ANN can predict the correct class of the received signal even if the input signals have not been seen before, which allow the model to learn from training dataset and generalize the model to any received signal. Moreover, ANN is a non-linear model and hence can predict the nonlinear received signal better than the linear model. Finally, the ANN parallel processing and the appropriate simple structure are two important properties for realizing ANN on hardware.

Furthermore, we have implemented a cognitive radio solution, which offers flexibility between the available spectrum and SNR. This solution has the capability to balance

between LSE and the overall channel capacity under a very low SNR. It constructs optimal communication symbols, which compensate for the difference in data rates under various noise levels. In addition, the PBCCS system integrates the modulator and channel encoder through a cross-layer approach. The binary data is encoded into the appropriate waveform according to the selected glossary. Each binary word is assigned to the artificially constructed patterns. The transmitter selects the appropriate set of patterns that maximize LSE.

2.1.2 Contribution

In our previous work [27], we have experimentally investigated the performance of PBCCS in an additive white Gaussian noise (AWGN) channel by employing 2-level Daubechies-2 wavelet (DB2) as a discrete wavelet transformation (DWT) to preprocess the received signal. With regards to this, in the current work, learning the appropriate communication patterns for the cognitive receiver and the influence factors on the received symbols are being studied. The main contributions of this paper are in fourfold.

- We analyzed various DWT approaches, which have an influence on the recognition rate of the ANN.
- We studied the effect of using 4- and 5-level DWT, which reduce the size of the ANN.
- We analyzed ANN with various back-propagation learning algorithms, which have an influence on system performance as well as the speed of learning.
- Finally, we showed that the space complexity of the receiver exhibits a reduced ANN structure in terms of inputs and the hidden layer. As fewer resources were used, the receiver could be implemented with fewer hardware units.

Table 2.2 : General ANN usage survey within cognitive radio.

Ref.	Brief summary
[21]	Wavelet cyclic feature has been proposed to reduce the complexity of calculating classical cyclic spectrum and a FFNN has been used to classify the received signals into BPSK, QPSK, MSK and 2-FSK. **Cons:** limited to low-order modulation schemes.
[22]	Based on the instantaneous temporal features the authors have proposed a FFNN and probabilistic ANN to classify the received signals into 2- and 4-ASK, BPSK, QPSK, 2- and 4-FSK, 8-PSK, 16-QAM. **Cons:** Large ANN architecture and requires prior information on some specific parameters to guarantee the highest accuracy and reliable recognition.
[23]	Based on the instantaneous temporal features (the maximum value of the spectral power density, the standard deviation of the direct and absolute instantaneous phase and the standard deviation of the normalized instantaneous amplitude), the authors have proposed a FFNN to classify the received signals into five classes, namely; 2- and 4-ASK, BPSK and QPSK. **Cons:** low-order modulations were considered. This approach requires prior information on specific parameters to guarantee the highest accuracy and reliable recognition.
[25]	Based on the extracted CWT instantaneous features (the mean, variance and central moments values), the authors have proposed an ANN to classify the modulation scheme into k-ASK, k-PSK, k-FSK, k-QAM, OOK and MSK.
[26]	Based on the extracted instantaneous temporal features, the authors proposed a rule-based approach to discriminate between 15 modulation schemes. **Cons:** due to limited number of features and signal sensitivity, the approach was unable to classify the same modulation schemes of higher order.
[28]	FFNN, radial basis function ANN and multi-class support vector machine (SVM) have been suggested to classify the modulation technique of the received signal into 2- and 4-FSK, 4-ASK, 8-ASK,2-PSK, 4-PSK,8-PSK, V32, 8-, 16-, 32-, and 64-QAM. **Cons:** it requires prior information on specific parameters to guarantee the highest accuracy and reliable recognition.
[29]	An expert discrete wavelet adaptive network based on fuzzy inference system has been proposed for classifying the digital modulated signals into 8-ASK, 8-FSK, 8-PSK, and 8-QAM. **Cons:** very large ANN structure, with four hidden layers.
[30]	A system that is only based on wavelet transform has been developed, where a comparison between signals and templates in wavelet domain has been adapted to classify the received signals into 2-ASK, 2-FSK and BPSK. **Cons:** binary digital modulation schemes were considered. It also requires prior signal information, such as, carrier frequency and symbol duration.
[31]	A system that is based solely on DWT and signals statistics was used to classify the modulated received signals into 16-QAM, QPSK and BPSK. **Cons:** degradation of performance at SNR appeared when the ANN was trained on signals with lower SNR.

2.1.3 Paper organization

The rest of this paper is organized in the following way. Section 2.2 describes the structure of the PBCCS model and its blocks in detail. It also gives a short introduction on wavelet and neural networks. In section 2.3, we evaluate the performance of the PBCCS system. In section 2.4, we conclude the paper and recommend directions for future work.

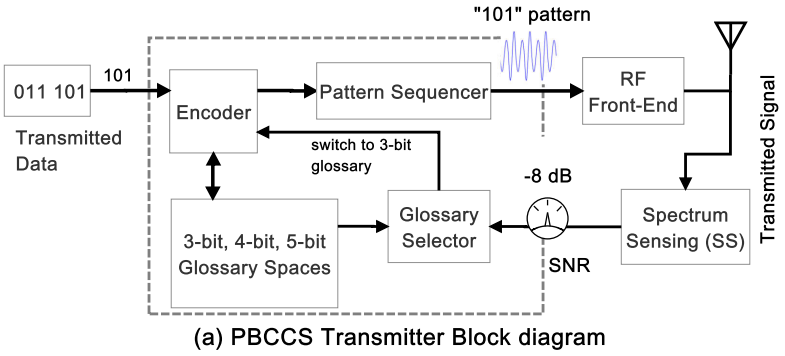
2.2 PBCCS Structure

The proposed system consists of two main parts, the transmitter and receiver. Basically, the system employs pattern-based encoding at the transmitter and a wavelet-preprocessed artificial neural network based decoder at the receiver. In this section, we describe the details of the individual parts of PBCCS.

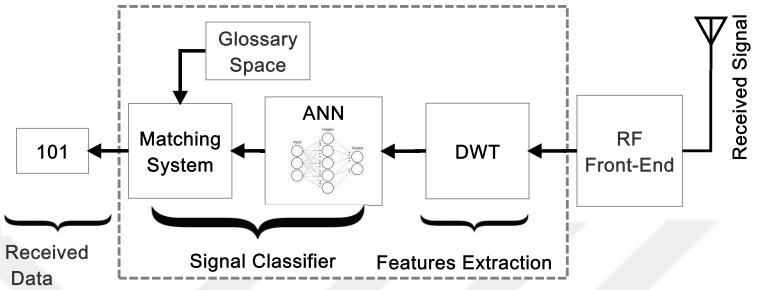
2.2.1 The transmitter of PBCCS

The transmitter of PBCCS is responsible for three tasks: 1) selecting the appropriate glossary with respect to the SNR level, 2) encoding the user data and 3) transmitting the signal through the antenna. In PBCCS, the modulation is performed by using the Sinusoidal Pattern Envelope Construction (SPEC) algorithm [10]. The SPEC algorithm is used to prevent unwanted extra spectral usage, and it guarantees that the signal's pattern ends at its initial point to ensure a zero-power density in average and has no high-frequency components.

The SPEC algorithm has two essential parameters—namely, "depth" and "level". Depth determines the length of the pattern in terms of the time—i.e., number of periods. Meanwhile, level identifies a value for each feature of the signal pattern. It represents the maximum and the minimum values of any signal characteristics (amplitude (A), frequency (F) or phase (P) at depth i. All possible outcomes in the SPEC algorithm are due to the changes in the A, F and P features of the signal. Figure 2.1.a shows the block diagram of the transmitter, which consists of an encoder, a glossary space, and a glossary selector. The glossary is an information encoding method. It is composed of different patterns generated by the SPEC algorithm. The



(a) PBCCS Transmitter Block diagram



(b) PBCCS Receiver Block diagram

Figure 2.1 : The block diagram of the PBCCS. (a) The transmitter selects a 3-bit glossary space, so the corresponding "101" waveform is transmitted. (b) The receiver receives a distorted signal and produces the corresponding bits.

SPEC is responsible for combining different sinusoidal waveforms to form a symbol as shown in equation (2.1). The generated signal of each symbol has different waveforms that change over time in terms of amplitude, frequency and phase.

$$m_i(t) = \sum_{i=0}^{J} a_i * x_i(t) * cos(2 * \pi * f_i * t + \phi_i) \tag{2.1}$$

where $m_i(t)$ is the i^{th} pattern; a_i ,f_i and ϕ_i are the amplitude, frequency and phase of the signal, respectively; J is the total number of sinusoidal waves that the pattern contains (determined by the depth parameter); and $x_i(t)$ is defined as

$$x_i(t) = \begin{cases} 1 & \text{if } 2 * \pi * i \leq t < 2\pi * (i+1) + 2\pi, \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

Each block of k input data bits $d_i, i = 1, 2, ..., 2^k$ is mapped to pattern $m_i(t)$. All symbols, $m_i(t)$, are combined to form a k-bit glossary space. Figure 2.2 illustrates a 3-bit glossary space, containing eight signal patterns. In this work, the performance of 3-bit, 4-bit and 5-bit glossary spaces, with 8, 16 and 32 patterns, respectively, is studied.

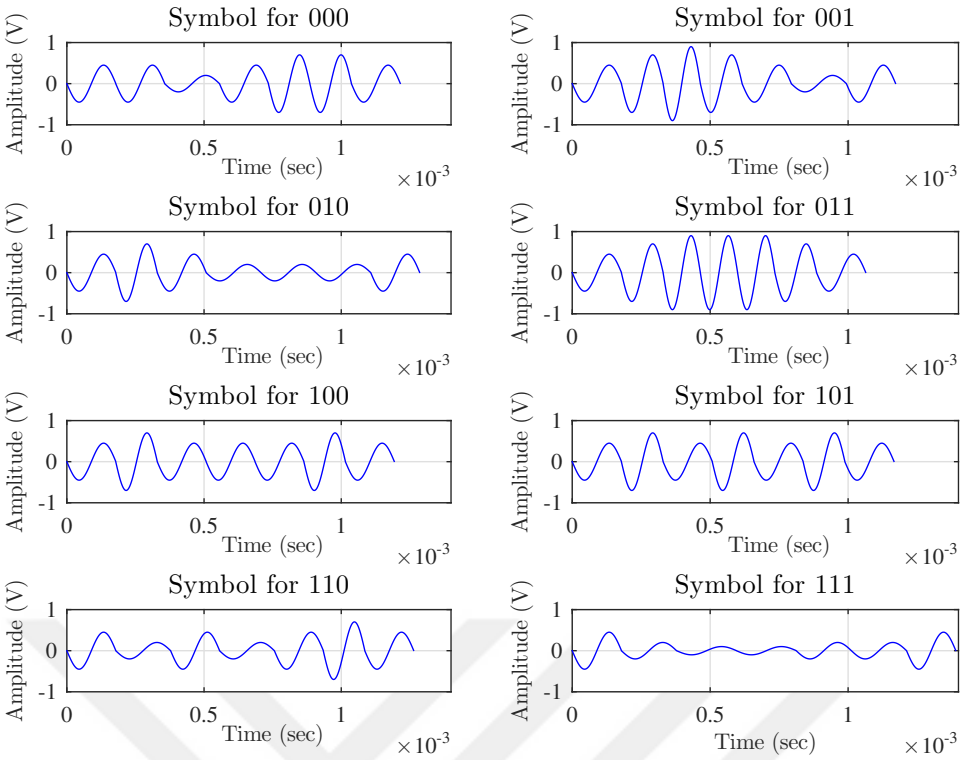


Figure 2.2 : The 3-bit glossary space with 5 levels, each of which has 7 patterns (depth is set to 6). The signal's amplitude is $[0.1, 0.2, 0.45, 0.7, 0.9]$ V, frequency is $[4.600, 5.000, 5.600, 6.600, 7.400]$ kHz, and phase shift is $[-\pi/2\ -\pi/4\ 0\ \pi/4\ \pi/2]$ rad.

The transmitted symbol, $m(t)$, contains a sequence of known data symbols, $m_1(t), m_2(t), \ldots$. According to the selected glossary, the pattern that matches the data index is selected and applied to the RF front-end.

The glossary selector is the core component in the transmitter's design, because it selects the most appropriate glossary from the glossary space as part of the adaptation process. It takes the glossary space information and channel spectral situation—i.e., the SNR value from the environment, as an input to determine the most proper glossaries set in the glossary space by computing the maximum likelihood value. For example, Figure 2.1 shows that the measured SNR is -8 dB. Therefore, the glossary selector switches to a 3-bit glossary and maps '101' to the sixth pattern (shown in Figure 2.2).

2.2.2 The receiver of PBCCS

The main modules of the receiver are the discrete wavelet transform unit and ANN, as illustrated in Figure 2.1.b. The aim of using ANN at the receiver is to predict the

original bits of the distorted received signal. The receiver does not construct a similar analog signal or estimate its parameter. Instead, it classifies the input samples to a known pattern, so that the correct bits can be inferred. In the following subsections, we briefly describe the functionality of each part of the receiver.

2.2.2.1 Features extraction and reduction

One of the aspects of signal classification is the selection of proper classification features. The goal of feature extraction is to obtain a set of features that can discriminate different received signals. In this work, the discrete wavelet transform (DWT) [32] is used to extract the signal features.

The discrete wavelet transform is a linear signal processing technique that transforms a signal $r(t)$ from the time domain to the "wavelet" domain—i.e., wavelet coefficients. A transformation from the time domain to the "wavelet" domain is analogous to the Fourier transform. The key difference between wavelet transform and Fourier transform is that wavelets are local in both time (via translation) and frequency (via dilation), whereas Fourier analysis is local only in frequency but not in time. Because the generated waveforms contain numerous nonstationary or transitory characteristics, which are often the most important parts of signals, Fourier analysis is unsuitable to describe such characteristics. Moreover, the received pattern signal can be represented by a compact form and hold most features that distinguish it from other patterns. As a result, the wavelet analysis is appropriate to capture the changes in the pattern's frequency over time and achieves better lossy compression, which dramatically reduces the size of ANN. In general, the received signal, $r(t)$, can be modeled in the AWGN channel as follows:

$$r(t) = m_s(t) + w_n(t) \tag{2.3}$$

where $m_s(t)$ denotes the original pattern signal, and $w_n(t)$ is white Gaussian noise with normal distribution. $r(t)$ is discretized in time so that $n-$point discrete signal $r[n]$ is constructed. The DWT is defined by equation (2.4) as follows:

$$W(j,k) = \sum_j \sum_k r(k)\psi_{jk}(n), \qquad j,k \in \mathbb{Z} \tag{2.4}$$

$$\psi_{jk}(n) = 2^{-j/2}\psi(2^{-j}n - k) \tag{2.5}$$

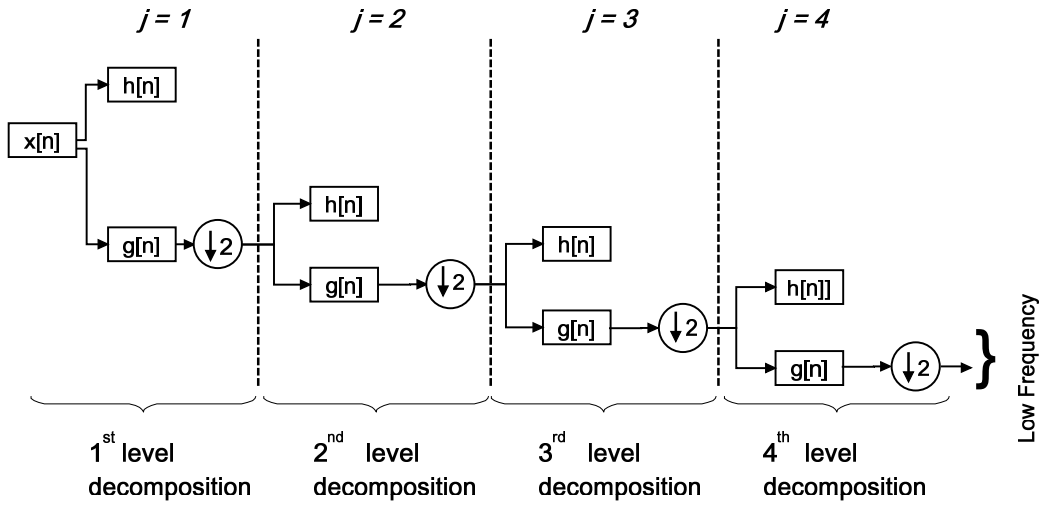


Figure 2.3 : The block diagram of a two-channel four-level DWT decomposition ($J = 4$) that decomposes a discrete signal into two parts.

where $W(j,k)$ are the wavelet transform coefficients; $\psi_{jk}(n)$ is the mother wavelet, and j and k are the scale parameter and shift parameter, respectively. In practice, it is inconvenient to apply equation (2.4) in calculating the coefficients. The DWT can be implemented as a series of high-pass and low-pass filters, called the recursive wavelet transform, which decomposes the signal $x[n]$ into two parts. The wavelet decomposition depends mainly on orthonormal filter banks. Figure 2.3 shows the signal decomposition by a two-channel wavelet structure, where $x[n]$ is the input signal, $h[n]$ is the high-pass filter, $g[n]$ is the low-pass filter, and $\downarrow 2$ is the down-sampling by a factor of two. In this way, each filter creates a series of coefficients that represent and compact the original information of the signal.

Mathematically, a signal $x[n]$ is composed of high and low frequencies as shown in equation (2.6). It shows that the obtained signal can be represented by using half the coefficients because they are decimated by 2.

$$x[k] = x_{high}[k-1] + x_{low}[k-1] \tag{2.6}$$

The filtered and decimated output of low-pass part is recursively passed through identical wavelet filter banks to add the dimension of varying resolution at every stage. Equations (2.7) and (2.8) are mathematical expressions of filtering a signal through a digital high-pass filter $h[n]$, and low-pass filter $g[n]$. This operation corresponds to convolution with an impulse response of $k-$tap filters.

$$y_{high}[k] = \sum_{n} x[n].h[2k-n] \tag{2.7}$$

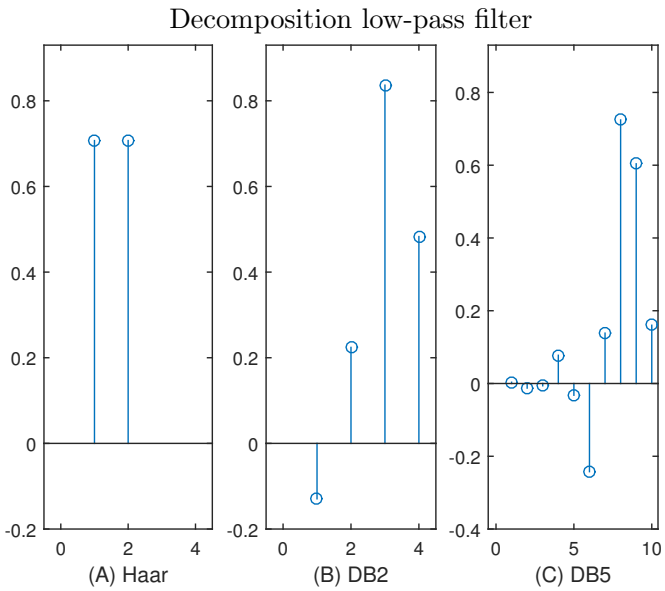


Figure 2.4 : Low-pass filter parameters of different wavelet families.

$$y_{low}[k] = \sum_{n} x[n].g[2k-n] \tag{2.8}$$

where n becomes $2n$ representing the down-sampling process. The output of the low-pass filter, $y_{low}[k]$, provides approximation signal, whereas the output of the high-pass filter, $y_{high}[k]$, provides detailed signal. In addition, equations (2.7) and (2.8) show that using DWT can not only greatly reduce the number of input nodes, but also effectively expresses the features of the received signal, thereby enhancing the ability of neural networks to recognize the signal.

A variety of different wavelet families has been proposed in the literature. In this work, the simplest wavelet family—i.e., the Haar wavelet and triangle wavelet are selected [32]. In addition, the 4- and 10-coefficient wavelet family (the second and the fifth orders of Daubechies wavelet—i.e., DB2, DB5), and the 6-coefficient wavelet family coiflets (C6) proposed by Daubechies are used [33]. The decomposition low-pass filter parameters of the Haar, DB2 and DB5 wavelet are shown in Figure 2.4.

2.2.2.2 Recognition layer

After extracting the proper features from the received signal, classifying these patterns into appropriate classes is the final step to recognize the symbol.

In this work, the artificial neural network (ANN) [34,35] is considered as a recognition layer to recover the transmitted data, and it forms the cognitive part of the PBCCS receiver. The main reason for choosing the preprocessed wavelet with ANN is its

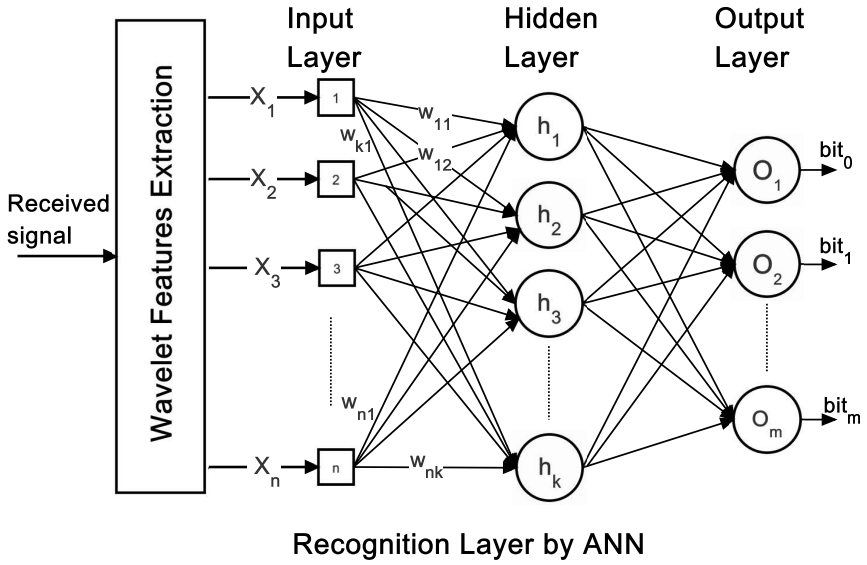


Figure 2.5 : A Feed-forward ANN with one hidden layer and an output layer. The extracted features are fed into ANN with n input nodes, and k and m are the number of hidden and output neurons, respectively. The output bits are the bits that were recognized by the FFNN.

high sensitivity for recognizing the amplitude, frequency and phase changes in the communication signal. The output that is produced by the ANN is decoded in the final stage into the correct bit sequence, as shown in Figure 2.5.

Also, in this work, the most common ANN model, namely multilayer perceptron (MLP), is used. MLP is a type of feed-forward neural network (FFNN) model that maps the input data onto a set of appropriate outputs. It consists of at least three layers—i.e., the input layer, one or more hidden layers and an output layer. The network is fully connected from one layer to the next as a directed acyclic graph (Figure 2.5). Each neuron is capable of multiplying the inputs by its weight and sum up the results. In other words, the neuron operations are performed by multipliers and adders.

Mathematically, for n arbitrary distinct received samples (x_i, t_i), where x_i is the extracted features' vector from the received signal, $x_i = [x_{i1}, x_{i2}, ..., x_{in}]^T \in \mathbb{R}^n$ and t_i is the target vector, $t_i = [t_{i1}, t_{i2}, ..., t_{im}]^T \in \mathbb{R}^m$. n and m are the size of the input feature and the target vectors, respectively. The target vector t_i represents the actual sequence of bits that the recognition layer must produce.

A single hidden layer of a FFNN (Figure 2.5) with an activation function, $g(.)$ and k hidden neurons is mathematically modeled as

$$g(\sum_{i=1}^{n} w_{ji} \cdot x_i + b_j) = y_j, \quad j = 1,2,...,k \tag{2.9}$$

where $w_{ji} = [w_{j1}, w_{j2}, ..., w_{jk}]^T$ is the weight vector connecting the j^{th} hidden neurons with the inputs and b_j is the bias value of the j^{th} hidden neuron. The bias allows the sigmoid function curve to be shifted horizontally along the input axis while leaving the shape of the function unchanged. $w_{ji} \cdot x_i$ denotes the inner product of w_{ji} and x_j.

The result of the j^{th} output neuron can be mathematically computed as shown in equation (2.10):

$$g(\sum_{i=1}^{k} \beta_{ji} \cdot y_i + b_j) = O_j, \quad j = 1,2,...,M \tag{2.10}$$

where M is the total number of output neurons. $\beta_j i = [\beta_{j1}, \beta_{j2}, ..., \beta_{jm}]^T$ is the weight vector connecting the j^{th} hidden neurons and output neurons and b_j is the bias value of the j^{th} output neuron.

Because the final goal is to find the relation between the input x_i and the target t_i, the activation function $g(.)$ can approximate these n received samples with zero mean error, such that $\sum_{j=1}^{N} \|t_i - y_i\| = 0$. Thus, there exist β_i, w_i and b_i such that

$$\sum_{i=1}^{n} \beta_i g(w_i \cdot x_j + b_i) = t_j, \quad j = 1,2,...,m \tag{2.11}$$

The back-propagation algorithm (BP) [34] is used to compute the weights and biases of the ANN by minimizing the error function in weight space using gradient descent.

The received sequence of bits, $\hat{a} = \hat{a_1}, \hat{a_2}, \ldots, \hat{a_m}$ is approximated from the output neurons as shown in equation (2.12):

$$\hat{a_j} = \begin{cases} 1 & \text{if } O_j \geq 0.5, \\ 0 & \text{otherwise} \end{cases} \tag{2.12}$$

where O_j is the j^{th} the output neuron.

As a final point, the structure of the FFNN should be modular and simple, so that the hardware architecture can be efficiently realized on floating point digital signal processor (DSP) or a field programmable gate array (FPGA). As there are various combinations of designing a FFNN, an efficient design should include the following aspects

- Input Neurons: the number of neurons is equivalent to the sample size of the DWT vector.
- Hidden Neurons: due to low complexity and high applicability perspective [10], a single hidden layer with few number of neurons should be used. The number of neurons will be determined by the cross validation method.
- Output Neurons: The number of output neurons should be identical to the size of the glossary space—i.e., total number of patterns. However, each glossary differs in the number of symbols that it represents. For instance, the $3-$bit glossary has 8 symbols while $4-$bit glossary has 16 symbols. Therefore, the size of the glossary can be added to determine the width of the output sequence bits.

2.3 Experimental Results

In this section, the performance of the proposed PBCCS based on the extracted features (discussed in Section 2.2) is verified in an AWGN channel. At the end of this section, the proposed approach complexity is presented.

2.3.1 Simulation settings

Simulations were carried out to transmit 2^k different symbols (k-bit glossary) at various SNR levels. The SNR levels were in the range of $[-15, 25]$ dB. The ANN parameters are shown in Table 2.4. All experiments were performed using Matlab software.

2.3.2 Constructing the glossary

Each test pattern is constructed with five sub-signal. According to the SPEC algorithm, patterns are generated with bandwidth-limited switching among different frequency, phase and amplitude levels. The features of the communication signal (frequency, phase and amplitude) are chosen from five different levels, listed in Table 2.3. The depth is set to 6, which indicates that each signal symbol is constructed from 7 sub-patterns.

Synthetic signals that represent the symbols was validated on hardware. For this purpose, we used an FMC150 [36] daughter card attached to a Xilinx ML605 board [37]. The FMC150 is a dual 14-bit channel ADC and dual channel 16-bit DAC FMC

daughter card. Figure 2.6 shows the synthesized signal by the hardware against a simulated signal generated by Matlab. There are 7 sub-patterns in each signal. The frequencies of each pattern are 1.25, 5, 6.25, 5, 1.25, 2.5 and 1.25 MHz. The amplitude and phase are identical to the values presented in Table 2.3. The frequency of the simulated signal might slightly differ from the real one owing to the register precision of the FMC150—i.e., the measured frequency of each sub-pattern is 1.229, 4.9020, 6.1728, 4.9020, 1.229, 2.457 and 1.229 MHz. The mean voltage of the baseband signal is $2.9710 * 10^{-4}$, which is approximately zero.

Table 2.3 : The based-signal specifications that were used to construct the glossary.

Feature	-L2	-L1	L0	L1	L2
Frequency (kHz)	4.600	5.000	5.600	6.600	7.400
Phase (rad)	$-\pi/2$	$-\pi/4$	0	$\pi/4$	$\pi/2$
Amplitude (V)	0.1	0.2	0.45	0.7	0.9

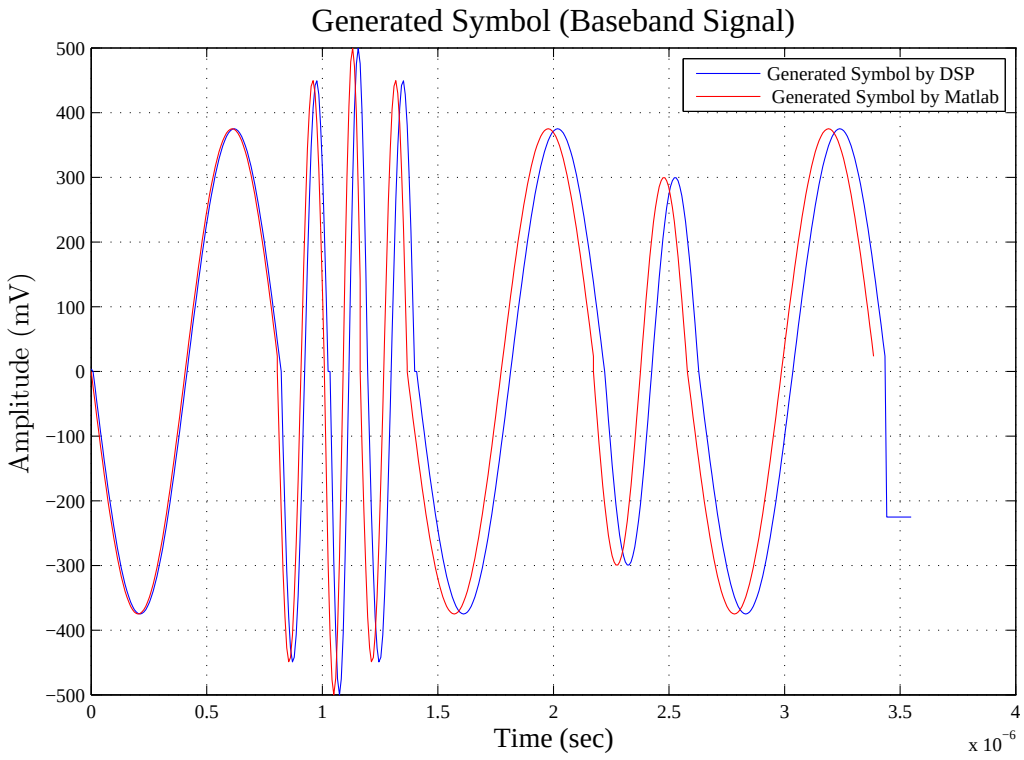


Figure 2.6 : Synthetic signal (baseband signal) by using Matlab compared with the real synthetic signal generated by the hardware. $F_s = 122.88$MHz.

2.3.3 Learning process and model evaluation

Before assessing the system, two datasets were generated for each symbol. These sets should be used for learning (training) and testing (evaluating). The learning set is used to derive the model offline, whereas the test set is used to estimate model's accuracy online, as shown in Figure 2.7. Symbols that were used during the learning stage would not be involved in the testing stage. Moreover, the learning dataset should be larger than the testing dataset, so that the model can be built from a large space of sampled signals. For example, the dataset of the 3-bit glossary contains $248,000$ symbols and $74,400$ symbols for learning and testing, respectively. In other words, 70% of the total symbols were used for learning.

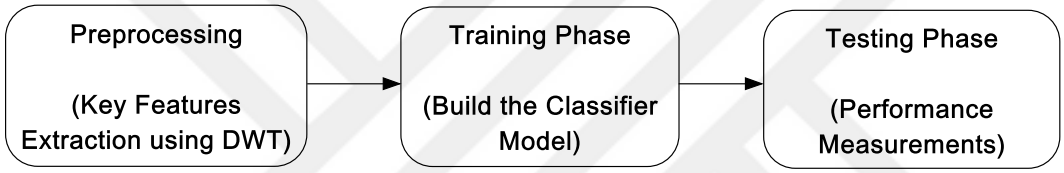


Figure 2.7 : The functional blocks of the experiments.

Table 2.4 : ANN settings.

Parameters	value
Number of Layers	2
Number of input nodes	$220,55,27$
Number of output neurons	$3,4,5$
Activation function	Sigmoid
Training Algorithm	Scaled Conjugate Gradient
Gradient Error level	1×10^{-6}
Performance Function	Mean Squared Error
Number of neurons in hidden layer	Varied from 8 to 40
Maximum number of epochs to train	3000

To assess the accuracy of the wavelet filter banks and ANN model, we use the success rate of the recognition symbol as an indication of the effectiveness of the receiver to correctly recognize the received symbols. It indicates the capability of the trained model to classify the received symbols in the testing set, which were not seen before. It also expresses the probability of correct classification, which is computed as follows:

$$Success\ Rate = \frac{1}{N}\Sigma_i^N x_i * 100 \tag{2.13}$$

where N is the total number of test symbols and x_i is an indicator whose value is 1 if the i^{th} symbol is correctly received. In other words, *Success Rate* measures the symbol error rate (SER).

In addition to the success rate, the BER performance is used to assess the accuracy of the whole system. It expresses the number of bit errors per second divided by the total number of transferred bits.

2.3.4 Classification performance of different wavelet families

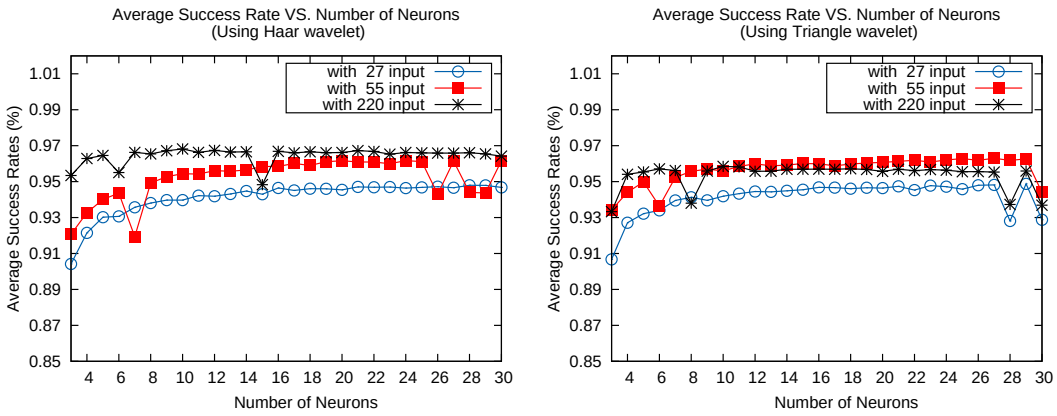
In this section, we study the effect of applying different wavelet families on the performance of the receiver for 3-bit glossary. The received signal has 880 samples. The number of input nodes was subsequently reduced from 220 to 55 and 27 to identify the most relevant input features to ANN by employing 2-, 4- and 5-level DWT decomposition ($j = 2, 4$ and 5). The ANN model was then trained for different numbers of neurons in the hidden layer. These experiments were repeated 10 times, and the success rate was averaged.

Figure 2.8.a illustrates the effect of using the Haar wavelet for various numbers of neurons in the input and hidden layers. It shows that the average success rate of the model is more than 90% for all scenarios. By using 27 neurons at the input layer, the success rate is greater than 94% with 12 neurons or more at the hidden layer.

In Figure 2.8.b, the effect of using a triangle wavelet is shown, which is similar to Figure 2.8.a. However, the success rate of the model with 55 neurons outperforms the model that has 220 inputs. The improvement of the input size reduction to 55 inputs by using DB2 is illustrated in Figure 2.8.c. The success rate is greater than 97% with more than 12 hidden neurons (with the exception of the case with 27 inputs).

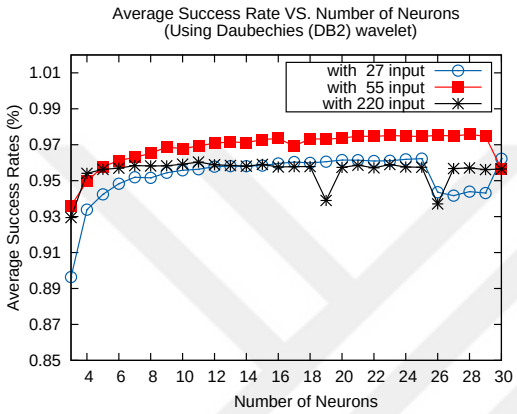
DB5 wavelet has similar performance compared to the DB2, as shown in Figure 2.8.d. Similar result was also obtained by using a coiflets (C6) wavelet as shown in Figure 2.8.e. The figure shows that with 55 input nodes and 10 hidden nodes, the performance is improved compared with the experiment of 220 input nodes.

To analyze the previous results, the success rates of PBCCS are elaborated in the $[-20, 15]$ dB range for various network configurations in the hidden layer (Figure 2.9). When the SNR is equal to or greater than -12 dB, the success rate is greater

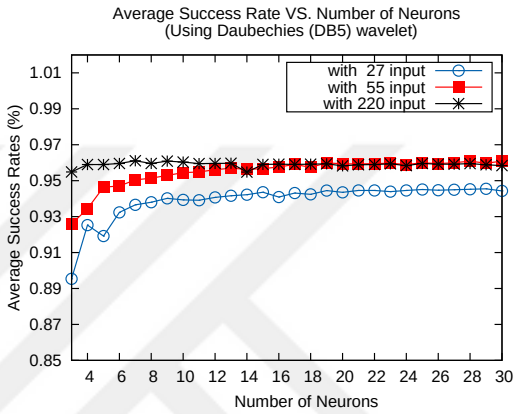


(a) Haar wavelet

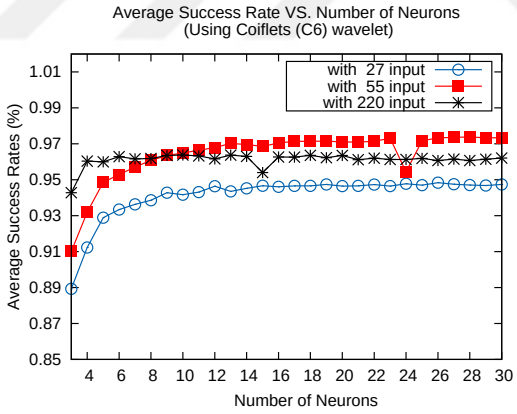
(b) Triangle wavelet



(c) Daubechies (DB2) wavelet



(d) Daubechies (DB5) wavelet

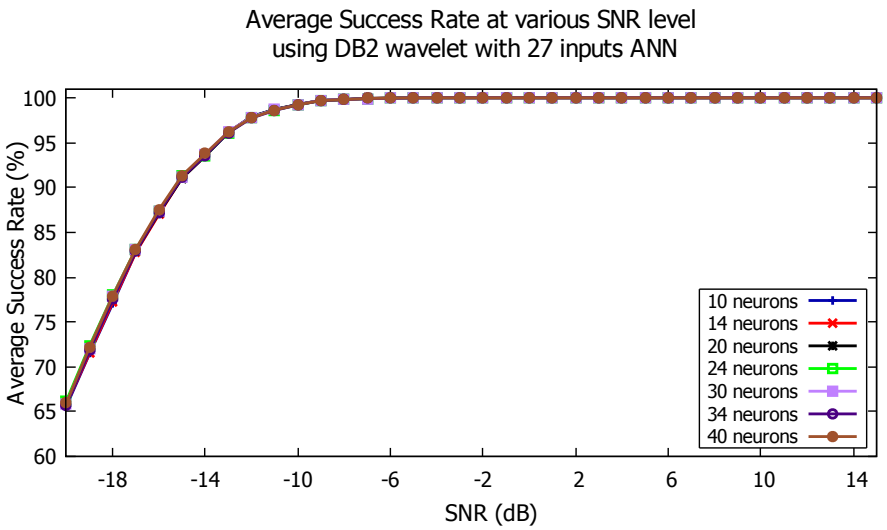


(e) Coiflets (C6) wavelet

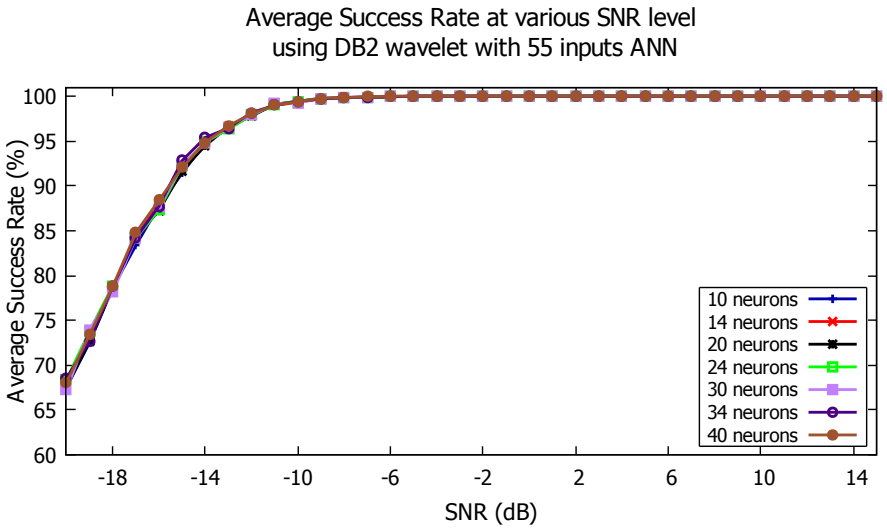
Figure 2.8 : The probability of correct classification (average success rate) vs. number of hidden neurons by using different wavelet transforms and various input size in an AWGN channel.

than 99.0% for all configurations, which means that there is no error in the received symbols ($SER = 0.1$). Below -12 dB, the success rate linearly decreases to 65%. In all cases, using 10 hidden neurons or more layer does not improve the success rate.

In summary, we found that for 3-bit glossary the DB2 wavelet has better performance compared with other wavelet families studied in our tests. It is also found that with



(a) Using 5-Level DB2 DWT (27 coefficients)



(b) Using 4-Level DB2 DWT (55 coefficients)

Figure 2.9 : The probability of correct classification (success rates) in AWGN channel when applying a DB2 wavelet connected to FFNN with 10, 14, 20, 24, 30, 34 and 40 hidden neurons.

a $27-$input ANN, the performance is better than that when using many extracted features. Furthermore, the use of 14 hidden neurons or more has a similar recognition rate to a network that contains 8 hidden neurons. As a result, an ANN that is based on 5-level DWT can be realized with 27 inputs and as minimum as 14 hidden neurons. This reduction will use fewer resources during the hardware design realization.

2.3.5 Classification performance of various learning algorithms

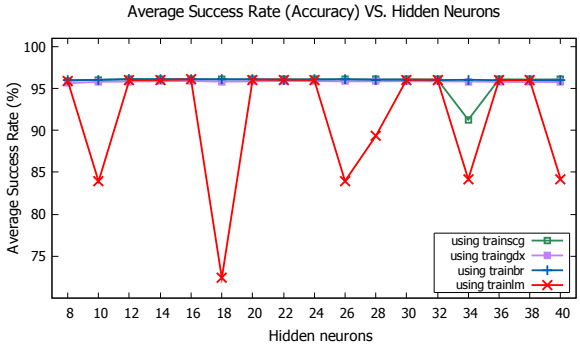
In this section, we examined 4 popular back-propagation learning algorithms—namely, (Levenberg- Marquardt (LM), scaled conjugate gradient (SCG), gradient descent with momentum and adaptive learning rate back-propagation (GDX) and Bayesian regularization back-propagation (BR) in terms of speed and the number of iterations to achieve the same performance goal (MSE)=0.01. The ANN parameters are set according to Table 2.4. For LM algorithm, the maximum number of iterations was set to 25 to limit the learning time as shown in Table 2.5. It is worth mentioning that we choose the best parameters for LM algorithms by experimental study.

Table 2.5 : Learning algorithms' settings.

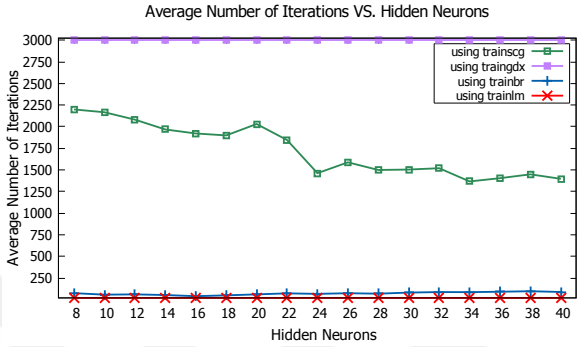
Parameter	value
Maximum number of epochs to train	25
Initial μ value	0.1
Decrease factor (α)	0.2
Increase factor (β)	6
Minimum performance gradient	10^{-5}
Performance Function	Mean Squared Error

The average training accuracies of each algorithms over various ANN structures are shown in Figure 2.10.a. It indicates that all learning algorithms achieve 96%. The LM algorithm is failed in some experiments because the algorithm reaches the maximum number of iterations (Figure 2.10).b. Figure 2.10.c shows that as the number of hidden units increases, the learning time of both LM and BR algorithms is significantly increased, whereas the learning time of both SCG and GDX algorithms is constant and less than 20 minutes. The average learning time of BR and LM with 40 hidden units network are 3 and 2 hours, while the number of iterations is less than 25 and 100 iterations, which means that both algorithms have high computation complexity.

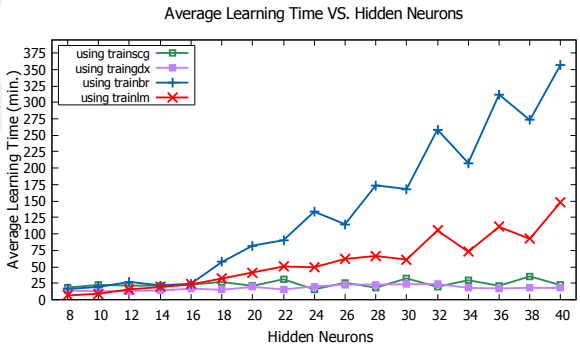
The average MSE versus number of hidden neurons are shown in Figure 2.10.d. Smaller values that are close to zero are better because they indicate that the MLP had fitted the data well. SCG, GDX and BR learning algorithms have better performance compared with LM algorithm because the number of iterations of LM was limited to 25 iterations. Increasing the number iterations will improve the MSE values but has dramatically effect on learning time.



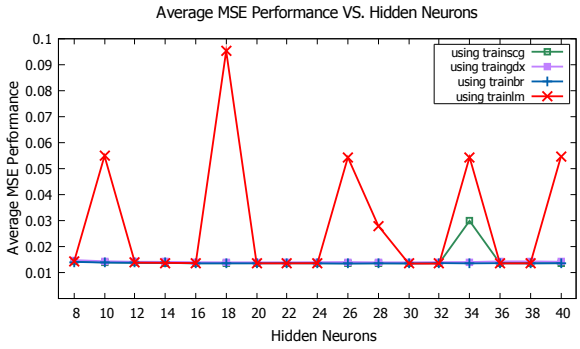
(a) Average success rate (Accuracy)



(b) Average number of iterations



(c) Average learning time



(d) Average MSE Performance

Figure 2.10 : Comparison between SCG, LM, GDX and BR learning algorithms with respect to a) Accuracy. b) Number of iterations. c) Learning time., and d) MSE Performance.

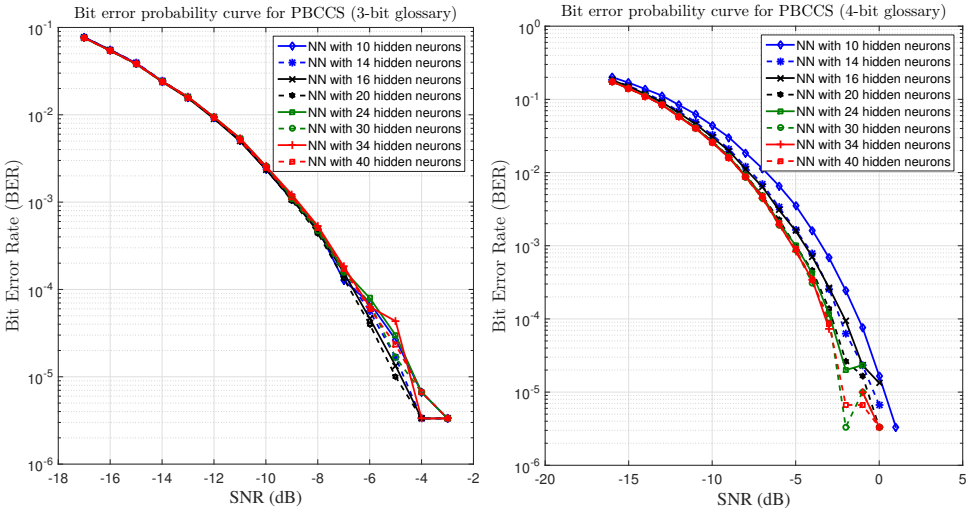
2.3.6 System performance with k-bit glossary spaces in AWGN channel

Based on the previous results, DB2 wavelet decomposition was applied to extract 27 features from the received signal. We simulated the BER by applying $3 * 10^5$ at each SNR level and measuring the number of uncorrected received bits. In Figure 2.11, the SNR curves illustrate the 3-, 4- and 5-bit glossary performance with various hidden neurons. The total area under each curve indicates the overall system performance under different noise and data bit error rate (BER) levels. It means that the carve with minimum area has a better performance. It is depicted in Figure 2.11.a that the BER is 10^{-5} at -5 dB for most ANN configurations. Similarly, Figure 2.11.b presents that the curve for 20 neurons has SNR of approximately -2 dB at 10^{-5}. Also, it shows that the system performance improves with increasing number of neurons due to the fact that the recognition capability could be enhanced as the number of hidden neurons is increased. Figure 2.11.c shows similar performance as shown in Figure 2.11.b except that SNR is approximately 4 dB at 10^{-5}. Figure 2.11 shows a strange behavior as the BER carve reaches 10^{-5}, where some errors increases again. We expect that the ANN could not distinguish between the samples either because the learning process is not enough or it overfits the data.

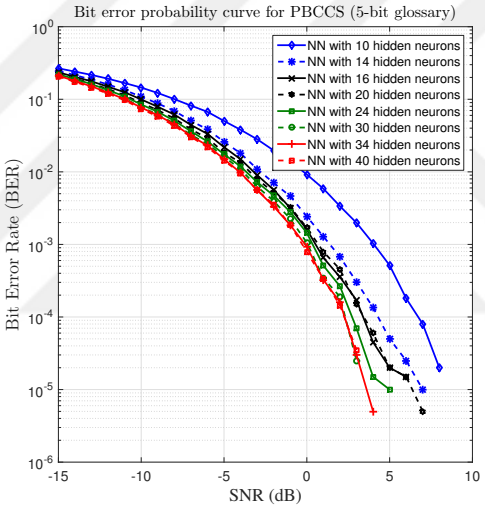
In summary, we found that the best performance of 4- and 5-bit glossary space is achieved when the number of hidden neurons is between 20 and 40 neurons. This means that the BER of an ANN with a hidden layer of 20 neurons is approximately equal to an ANN configuration with higher hidden units. This phenomenon indicates that an increasing number of hidden neurons does not always improve the performance. As a result, the best PBCCS performance can be realized with a fixed-structure ANN of 27 inputs nodes and 20 hidden neurons for 3-, 4- and 5-bit glossaries.

2.3.7 Spectral efficiency

The ability of the system to balance between the spectral efficiency under a very low SNR is an advantage of the proposed scheme. For each glossary, we construct a random signal of 2000 symbols to measure the average data rate and the occupied bandwidth. Table 2.6 shows the spectral efficiency, η, of the previous constructed glossaries.



(a) with 3-bit glossary (b) with 4-bit glossary



(c) with 5-bit glossary

Figure 2.11 : Bit error rate for 3 different glossaries in an AWGN channel. Each curve represents the number of neurons at the hidden layer.

Table 2.6 : Spectral efficiency of PBCCS at $P_e = 10^{-5}$ bit error probability.

Glossary	3-bit	4-bit	5-bit
M (symbols)	8	16	32
SNR value (dB)	-5	-2	4
Data Rate (kbps)	2.508	3.553	4.6
Average BW (kHz)	0.960	0.881	0.9135
LSE,η, (bps/Hz)	2.61	4.0318	5.03

The most significant feature of the proposed method is its competence to operate in a region beyond the Shannon boundary. The Shannon boundary is drawn in Figure 2.12 by using the Shannon limit theorem (equation (6.5—49), [2]), which is rewritten as follows:

$$\frac{E_b}{N_0} > \frac{2^{\eta}-1}{\eta} \tag{2.14}$$

where

$$\eta < \log_2(1+\frac{E_b R}{N_0 W}) = \log_2(1+\eta\frac{E_b}{N_0}) \tag{2.15}$$

Equation 2.14 states the condition of reliable communication in terms of bandwidth efficiency, η, and power efficiency in terms of E_b/N_0. Figure 2.12 shows the minimum value of $E_b/N_0 = -1.59$ for which reliable communication is possible. The marks in the figure show the best working point regarding the glossary at a 10^{-5} bit error rate and $M-$QAM. This figure depicts that the 4-bit glossary and 16-QAM have the same spectral efficiency, but the 4-bit glossary is more robust to high levels of AWGN than 16-QAM. Similarly, 5-bit glossary is more robust than $32-$QAM at the same spectral efficiency, $\eta = 5$.

However, the most interesting finding is that the $3-$bit glossary breaks down the Shannon limits. This phenomenon occurs because the ANN can perfectly discriminate among the eight signals of the $3-$bit glossary. In the learning step, the ANN model has constructed a model that allows the PBCCS to classify any unseen signal, in a manner analogous to memory. Another reason for breaking down the Shannon limit is the usage of non-periodic signals. Shannon's law is restricted to periodic signals [38]. The PBCCS constructs a non-periodic and uncorrelated communication waveforms that provide a manageable SNR capability between high noise redundancy and high data bandwidth requirements under observed spectrum conditions.

It is worth mentioning that this result depends on the difference between the bandwidth of recovered digital data based on a priori information in the glossary and the raw physical data bandwidth inside the communication medium. In addition, the synchronization overhead between the transmitted symbols is not considered.

2.3.8 Bit error rate comparison

A comparative analysis between the PBCCS and of the matched filter is shown in Figure 2.13. The total area under each curve indicates the overall system performance

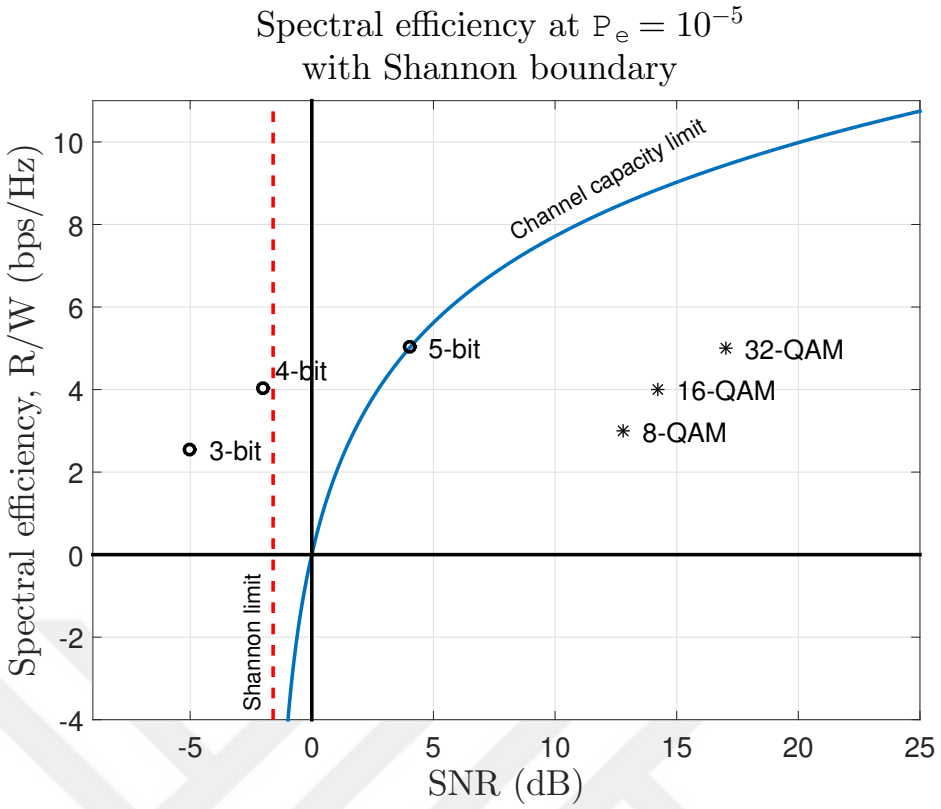


Figure 2.12 : Comparison of PBCCS and m-QAM in an AWGN channel at $P_e = 10^{-5}$ bit error probability. The neural network contains 20 hidden neurons.

under different noise and data BER levels. According to Figure 2.13, $3-,4-,5$-bit glossary constructed by PBCCS modulation have 20 dB, 21 dB and 22 dB better performances at 10^{-5} BER than 8-QAM, 16-QAM, and 32-QAM, respectively. It is worth noting that each k-bit symbol space has 2^k symbols as m-QAM because $m = 2^k$. Also, $3-,4-,5$-bit glossary outperforms the $4-,8-,16$-PSK by $16,17$ and 23 dB.

The bit error probability for 16-QAM is given by (equation $(4.2—85)$, [2]):

$$P_{b_{16-QAM}} = 3Q(\sqrt{\frac{4E_b}{5N_o}}) - \frac{9}{4}[Q(\sqrt{\frac{4E_b}{5N_o}})]^2 \tag{2.16}$$

where $Q(x)$ is the Q-function. $Q(.)$ is a monotonically decreasing function of its argument; the probability of error decreases as the ratio $\frac{4E_b}{5N_o}$ increases. This means that the decision boundary of the QAM technique depends on increasing the signal energy, which makes the binary signals dissimilar. However, the PBCCS depends not only on the signal energy but also on the extracted features from the received signal by DWT and the nonlinear model of ANN, which enable the PBCCS model to reduce the probability of error.

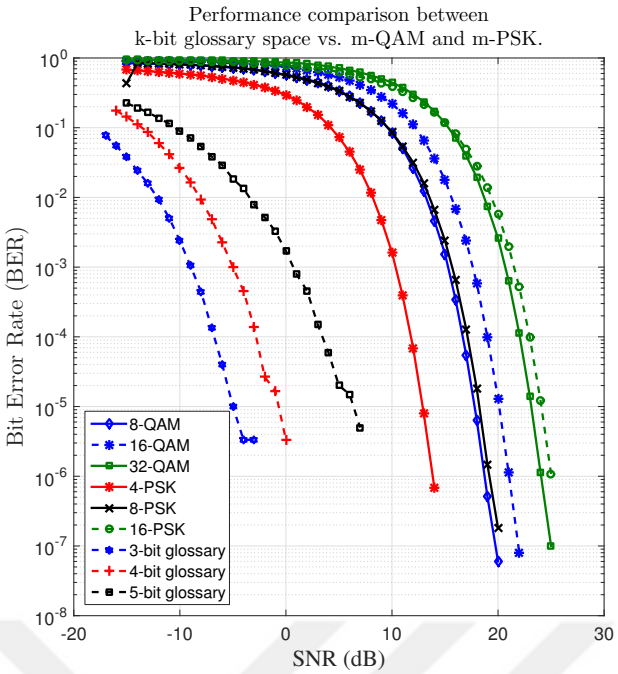


Figure 2.13 : The simulated performance of PBCCS and *m*-QAM in an AWGN channel. The neural network contains 20 hidden neurons.

2.3.9 System performance comparison with AMC

Because this work and similar works have used different methodologies in signal type recognition, direct comparison with these works is difficult. Table 2.7 compares the PBCCS with some of AMC techniques. A brief summary of each recognition approach was given in Table 2.2. It is clear that the proposed PBCCS outperforms the previous approaches (at -11 dB) because the symbols that are stored in the glossary was designed with properties that allow the ANN to discriminate between them.

2.3.10 Receiver space complexity

After simulating the proposed approach, the next step is to verify the design on real hardware such as a field-programmable gate array (FPGA) and application-specific integrated circuit (ASIC). We prefer using FPGA because the parallel structure of an ANN and the similarity of neurons makes its design simple and straightforward.

Each FPGA comes with limited resources, which poses challenges for real implementation. Space complexity gives an indication of the number of used functional units. It approximates the numbers of connections, multipliers and adders that the real design will occupy when it is implemented on an FPGA.

Table 2.7 : Comparison between different works in terms of features, ANN model and the achieved SNR with the recognition accuracy.

Ref.	Application	Applied Features	Type of ANN	Recognition accuracy (%)
[22]	AMC	Instantaneous temporal feature-based	FANN (5,19,8) and PANN (5,1800,8)	Overall success rate at -5 db are 65.63% and 55.5% respectively.
[23]	AMC	Instantaneous temporal feature-based modulation	FANN (4,7,5)	the overall success rate at -5 dB is 99.65%.
[25]	AMC	Continuous wavelet transform (CWT)	N/A	The overall success rate at 0 dB is 99.6% (using 10 features).
[26]	AMC	Instantaneous information and signal spectrum	N/A	The overall success rate at 3 dB is 98.6% (using 10 features).
[28]	AMC	Combination of the higher order moments, higher order cumulants and instantaneous characteristics of digital modulations	Radial basis function (RBF) probabilistic neural network (PNN)	The overall success rate at -3 dB is 87.50%. The overall success rate at -3 dB is 86.45%.
[29]	AMC	7-level DWT	Adaptive Network Based Fuzzy Inference Systems of 5 hidden layers	The overall success rate using DB2 at -5 dB is 98%.
[30]	AMC	Haar Wavelet Transform	N/A	The overall success rate at -7 dB is 99.71%.
[31]	AMC	Haar Wavelet Transform	N/A	The overall success rate at 5 dB is 97.93%.
[27]	Modulation classification and signal encoding	1-level DB2 DWT	FFNN(30,14,3)	The overall success rate at -11 dB for 3-bit glossaries is 96.0%.
PBCCS	Modulation classification and signal encoding	5-level DB2 DWT	FFNN(27,14,3), FFNN(27,14,4), FFNN(27,14,5)	The overall success rate at -11 dB for 3bit, 4-bit and 5-bit glossaries are 99.0%, 90.3% and 72.79%, respectively.

Table 2.8 : Space complexity comparison between the proposed approach and Time Delay ANN [10].

Parameters	Wavelet and ANN		Time Delay ANN	
	$k=3$	$k=4$	$k=3$	$k=4$
Number of Layers	2	2	2	2
bit size of glossary space	3	4	3	4
Input neurons	27	27	120	120
Hidden neurons	20	20	20	50
Output neurons	3	4	3	4
Number of multipliers	600	620	2460	2480

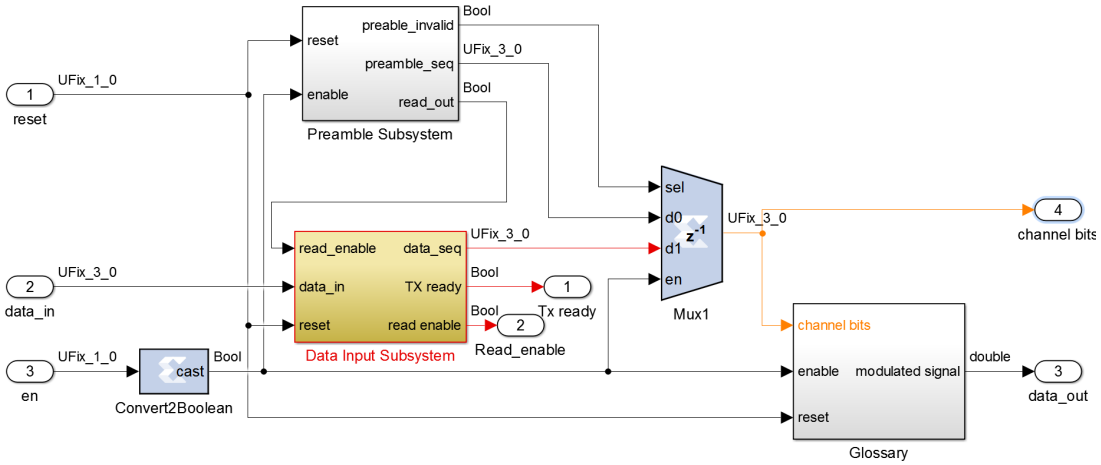


Figure 2.14 : The Simulink block diagram of the PBCCS transmitter.

The transmitter could be implemented by means of memory. Each glossary requires one memory (blocked RAM (BRAM)), which is indexed by the transmitted bits. For example, Xilinx Virtex VI xc6vlx240t-1ff1156 has 416 BRAMs, each storing 36 Kb [37]. Figure 2.14 shows the block diagram of the PBCCS transmitter, where the preamble was used for the synchronization. It shows only 3-bit glossary with 13×16 BRAM.

At the receiver, each neuron of the implemented ANN has a set of multipliers that are used to multiply the weight of the connections with the received data values. For example, if there are 26 2-input nodes, then each hidden neuron requires 26 multipliers and 26 2-input adders (including one adder to the bias). Because multipliers are more expensive than the adders and each FPGA comes with a limited number of them, they, in fact, significantly influence the design. For instance, Xilinx Virtex VI xc6vlx240t-1ff1156 has 768 multipliers (named DSP48E1) [37].

Table 2.8 shows a comparison between the architecture of the proposed approach compared with the one that is based on time delay ANN (TDNN) [10]. It shows that the total number of multipliers in the FFNN is drastically reduced compared with the TDNN. Moreover, the 3-bit glossary space requires $26 * 20 * 3 = 1560$ connections. This indicates the number of internal connections among the input, hidden and output layers of any used neural network, whereas TDNN requires $120 * 20 * 3 = 7200$ connections. In our approach, the number of units will be mainly affected by the size of the output layer, whereas in TDNN, the number of units will be affected by both, the size of the hidden and output layer.

In addition to the impact of the neural network, the wavelet decomposition has effect on the space complexity of the receiver. Similar to the finite impulse response (FIR) filter, the wavelet decomposition convolves wavelet coefficients with the received signals. This convolution process requires as many multiplication resources as the number of filter taps. For example, DB2 and DB5 can be implemented as 4-tap and 10-tap FIR filters, respectively. Furthermore, it is an iterative process—i.e., the output of one stage is an input to the next stage (Figure 2.3). The direct implementation DWT, known as the multiply-accumulate structure (MAC), requires as many resources as the number of stages times the numbers of filter taps—i.e., 5-level DB2 requires 20 multipliers.

An alternative but efficient implementation could be achieved by means of the distributed arithmetic algorithm (DAA) [39, 40]. DAA efficiently realizes the sum-of-products computation by means of memory (LUT), adders and shift registers, without employing any multipliers. That is, the total number of multipliers at the receiver will not be affected by the DWT implementation.

2.4 Conclusion

PBCCS was designed to increase the spectral efficiency by constructing a secure and non-periodic communication signals. In addition, PBCCS minimizes the bit error rate through optimized signal patterns that are decoded solely by DWT preprocessed signals and artificial neural network.

In this article, we analyzed the performance of ANN to recover original transmitted symbols using wavelets as a feature extractor. We applied different wavelet

decomposition techniques to study their effects. Several experiments were conducted to find the most appropriate wavelet family for PBCCS. The results obtained are intended to be a guidance tool in selecting the most appropriate operating point of the glossary selector with the discrete wavelet family at the receiver. We found out that the DB2 wavelet decomposition filter shows better performance compared with the other studied wavelet families. Thanks to DWT, a simple ANN structure was constructed with few hidden neurons, which is impossible for a third-party to predict. In addition, we studied the effect of various back-propagation learning algorithm. We could conclude that in terms of learning time and performance, both SCG and GDX are better to handle large dataset that includes thousands of signals. Finally, because of robustness to stationary noise, the proposed approach has a great advantage of less bit error, unlike the standard modulation techniques, which has higher bit error rate.

The simulation results also reveal that by using 5-level DWT and a neural network, SNR values of -5 dB, -2 dB and 4 dB are achieved at a BER of 10^{-5} for 3-bit, 4-bit and 5-bit glossary spaces, respectively. The advantage is obvious, because the transmitter can adapt the bit rate according to SNR values. Therefore, adaptive glossary and its performance can be considered in a future work.

An initial evaluation of hardware implementation was demonstrated, and the applicability of the proposed modulation technique and the recognition layer were discussed. In brief, according to our preliminary works on the FPGA platform, the system can be realized with limited level glossaries in the existing technology. The next future step of this work is validating the simulation and preliminary laboratory testbed results under real application and environmental conditions.

3. A COMPARATIVE ANALYSIS OF 1-LEVEL MULTIPLIER-FREE DISCRETE WAVELET TRANSFORM IMPLEMENTATIONS ON FPGAS[1]

3.1 Introduction

The Discrete Wavelet Transform (DWT) [41–43], a linear signal processing technique that transforms a signal from the time domain to the "wavelet" domain [32], employs different algorithms for decomposing a signal into an orthonormal time series with different frequency bands. The signal analysis can be performed using either Pyramid Algorithm (PA) [32] or Recursive Pyramid Transform (RPT) [44]. The PA algorithm is based on convolutions with quadrature mirror filters, which is not feasible for practical implementation. However, RPT decomposes the signal $x[n]$ into two parts using high- and low-pass filters, which can be implemented using filter banks [45].

1-D DWT has been implemented for signal de-noising, feature extraction and pattern recognition and compression in [42, 43, 46, 47]. The conventional convolution based DWT requires massive computations and consumes much area and power, which could be overcome by using the lifting-based scheme for the DWT, that was introduced by Sweldens [48]. Although, the lifting scheme is used to compute the output of low- and high-pass using fewer components, it may not be well suited for our application, owing to the pattern-based cognitive communication system (PBCCS's) nature, where the low frequencies components are much important than the higher ones [42]. Another advantage of using convolution-based DWT over lifting approach is that they do not require temporary registers to store the intermediate results and with appropriate design strategy, they could have better area and power efficiency [49, 50].

Rather implementing FIR filter via multipliers and an adder tree, a multiplier-free architecture is suggested due to their resultant in low complexity systems and their high throughput processing capability [40, 51, 52]. There are essentially two approaches

[1]This chapter is based on the paper "Alzaq, H. and Ustundag, B.B. (2018). A Comparative Analysis of 1-Level Multiplier-free Discrete Wavelet Transform Implementations on FPGAs, Turkish Journal of Electrical Engineering & Computer Science, 26(5): 1190-2004 "

for facilitating parallel processing, the Distributed Arithmetic Algorithm (DAA) and Residue Number System (RNS).

DAA efficiently performs the inner product function in a bit-serial manner via a Look-Up Table (LUT) scheme that is followed by shift accumulation operations of the LUT output [51]. The LUT, a memory element, stores precomputed partial results [40]. Alternatively, RNS is a highly parallel non-weighted arithmetic system that is based on the residue of division operation of integers using the LUT scheme [52]. Eventually, the RNS-based results are converted back to the equivalent binary number format using a Chinese Remainder Theorem (CRT) [53]. The key advantage of RNS is gained by reducing an arithmetic operation to a set of concurrent, but simple operations. Several applications, such as digital filters, benefit from the RNS design, as in [54] for example.

3.1.1 Contribution of this paper

In this paper, a three major 1-D DWT approaches are implemented on FPGA-based platforms and compared in terms of performance and energy requirements. The implementations are compared for different number of, multipliers, memory consumptions, number of taps (N) and levels (L) of the transform to show their advantages. To the best of our knowledge, no detailed comparisons of hardware implementations of the three major 1-D DWT design exist in literature. This comparison will give significant insight on which implementation is the most suitable for given values of relevant algorithmic parameters. Although there are many efficient design in literature, and we did not optimize the number of memories in any approach, so that we have a fair comparison.

The paper unfolds as follows. Section 3.2 reviews the theoretical background of DAA and RNS. Section 3 describes the implementation of discrete wavelet transform. We further show an analytical comparison between these approaches. Section 4 presents the performance results. Finally, the paper is concluded in Section 5.

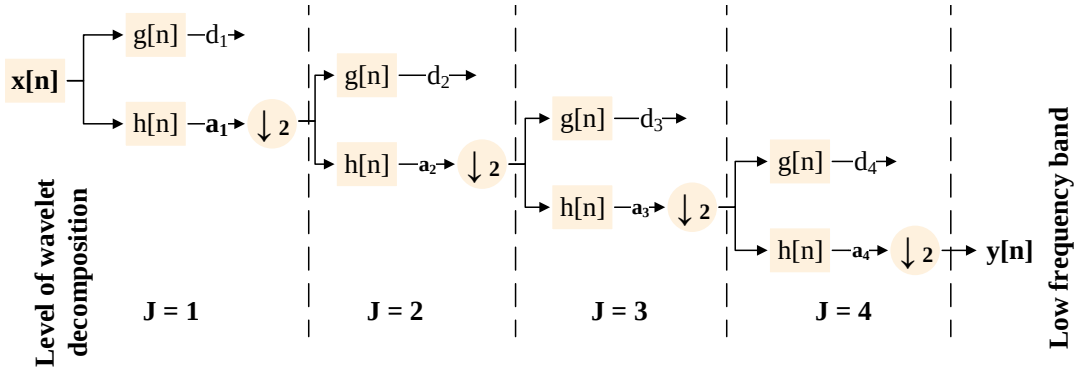


Figure 3.1 : Multi-level wavelet decomposition.

3.2 Background

3.2.1 Discrete wavelet transform

The wavelet decomposition mainly depends on the orthonormal filter banks. Figure 3.1 shows a two-channel wavelet structure for decomposition, where $x[n]$ is the input signal, $g[n]$ is the high-pass filter, $h[n]$ is the low-pass filter, and $\downarrow 2$ is the down-sampling by a factor of two. In this way, each filter creates a series of coefficients that represent and compact the original signal information.

Mathematically, a signal $y[n]$ consists of high and low frequency components, as shown in equation 3.1. It shows that the obtained signal can be represented by using half of the coefficients, because they are decimated by 2.

$$y[n] = y_{high}[n-1] + y_{low}[n-1] \tag{3.1}$$

The decimated low-pass filtered output is recursively passed through identical filter banks to add the dimension of varying resolution at every stage. Equation 3.2 mathematically expresses the filtering process of a signal through a digital low-pass filter $h[k]$. This operation corresponds to a convolution with an impulse response of k-tap filters.

$$y_{low}[n] = \sum_{k} h[k].x[2n-k] \tag{3.2}$$

where n becomes $2n$, representing the down-sampling process. The output $y_{low}[n]$ provides an approximation signal, while $y_{high}[n]$ provides the detailed signal.

3.2.2 Distributed arithmetic algorithm

Equation 3.2 shows that output y is the sum of multiplication of the filter coefficients and the input. The Distributed Arithmetic Algorithm (DAA) eliminates the need for hardware multipliers by performing the arithmetic operations in a bit serial computation [40]. Because the down sampling process follows each filter (as shown in Figure 3.1), equation 3.2 can be rewritten without the decimation factor as:

$$y_{low}[n] = \sum_{k=0}^{N-1} x[k].h[k] \tag{3.3}$$

Particularly, equation 3.3 is a computational intensive operation owing to multiplication of the real input values with the filter coefficients. Further simplification can be performed on the $x[k]$ in equation 3.3. Considering the representation of $x[k]$ as a fixed point arithmetic with length $L, x[k]$ becomes:

$$x[k] = -x[k]_0 + \sum_{l=1}^{L-1} x[k]_l.2^{-l} \tag{3.4}$$

where $x[k]_l$ is the l^{th} bit of $x[k]$ and$x[k]_0$ is the sign bit. Substituting equation 3.4 into 3.3, the output of the filter becomes:

$$y[n] = \left[\sum_{l=1}^{L-1} 2^{-l}.\sum_{k=0}^{N-1} h[k].x[k]_l\right] + \sum_{k=0}^{N-1} h[k](-x[k]_0) \tag{3.5}$$

Since $x[k]_l$ takes the value of either 0 or 1, $\sum_{k=0}^{N-1} h[k].x[k]_l$ may have only 2^N possible values. That is, rather than computing the summation at each iteration online, it is pre-computed and stored in a ROM, indexed by $x[k]_l$. In other words, equation 3.5 efficiently realizes the sum of product computation by memory (LUT), adders and shift register.

3.2.3 Residue number system (RNS)

The RNS [53] is a non-weighted number system that performs parallel carry-free addition and multiplication arithmetic. In DSP applications, which require intensive computations, the carry-free propagation allows for concurrent computation in each residue channel. The RNS moduli set, $P = m_1, m_2, \ldots, m_q$, consists of q channels. Each m_i represents a positive relatively prime integer, that is GCD$(m_i, m_j) = 1$, for

$i \neq j$.[2] Any number, $X \in \mathbb{Z}_M = 0, 1, \ldots M-1$, is uniquely represented in RNS by its residues $|X|_{m_i}$, which is the remainder of division X by m_i and M is defined in equation (3.6),

$$M = \Pi_{i=1}^{q} m_i = m_1 * m_2 * \cdots * m_q \tag{3.6}$$

where, M determines the range of unsigned numbers in $[0, M-1]$, and should be greater than the largest performed results. The implementation of RNS-based DWT is obtained from (3.3) as following,

$$y[n]_{m_i} = y_{m_i} = |(\sum_{k=0}^{N-1} |h[k]_{m_i}.x[n-k]_{m_i}|_{m_i})|_{m_i} \tag{3.7}$$

for each $m_i \in P$. This implies that a $q-$channel DWT is implemented by q FIR filters that are working in parallel.

Designing a robust RNS-based DWT requires choosing the moduli set and hardware design of residue to binary conversion. Most widely studied moduli sets are given as a power of two due to the attractive arithmetic properties of these modulo sets. For example, $\{2^n - 1, 2^n, 2^{n+1} - 1\}$ [55] and $\{2^n, 2^{2n} - 1, 2^{2n} + 1\}$ [56] have been investigated. In this work, the moduli set $P_n = \{2^n - 1, 2^n, 2^{n+1} - 1\}$ is used for three reasons. First, the multiplicative adder (MA) is simple and identical for $m_1 = 2^n - 1$ and $m_3 = 2^{n+1} - 1$. Second, for small $n = 7$, the dynamic range of P_7 is large, $M =$ 4145280, which could efficiently express real numbers in the range $[-2.5, 2.5]$ using 16-bit fixed-point representation, provided scaling and rounding are done properly. Third, the reverse converter unit is simple and regular [57] because it does not employ any ROM.

3.3 DWT Implementation Methodology

3.3.1 DWT implementation using DAA

DAA hides the explicit multiplications with a ROM lookup table. The memory stores all possible values of the inner product of a fixed w-bit with any possible combination of the DWT filter coefficients. The input data, $x[n]$, are signed fixed-point of 22-bit width, with 16 binary-point bits ($Q_{5,16}$). We assumed that the memory contents have

[2]The greatest common divisor (GCD) of two non-zero integers is the largest positive integer that divides them without a remainder.

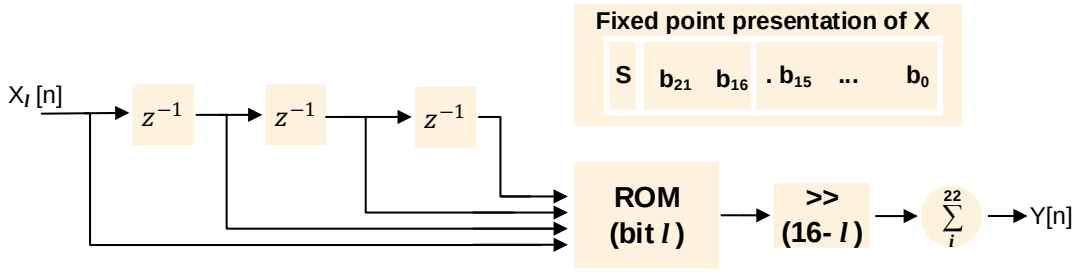


Figure 3.2 : The block diagram of DAA-based architecture of the DB2. For simplicity, we showed one ROM and one shift register.

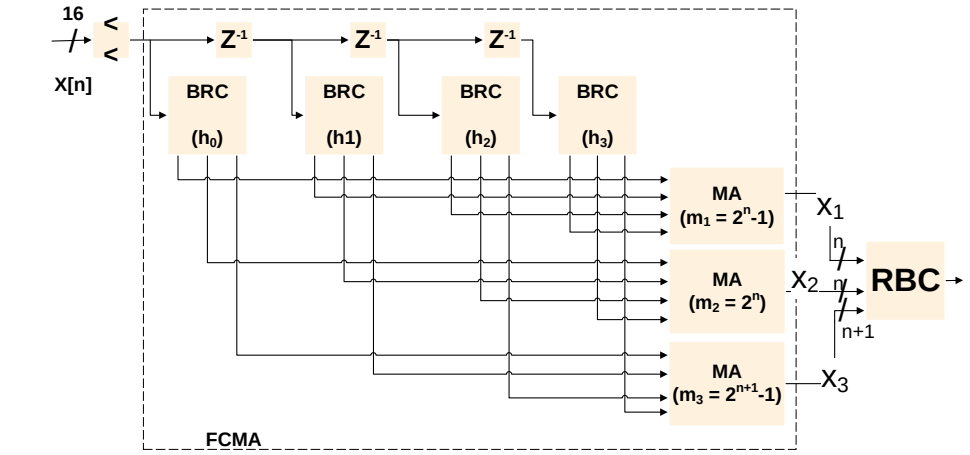
the same precision as the input, which is reasonable to give high enough accuracy for the fixed-point implementation. As a consequence, 22 ROMs, each consisting of 16 words, are required. Each ROM stores any possible combination of the four DWT filter coefficients, where the final result is a 22-bit signed fixed-point ($Q_{5,16}$). In order to decrease the number of memory, the width should be reduced, which will have impact on the output precision.

Figure 3.2 shows the block diagram of one bit DAA at position l. This block contains one ROM (4×22) and one shift register.Because the word's length w of the input x is $22-$bits, the actual design contains 22 blocks. In addition, 21 adders are required to sum up the partial results.

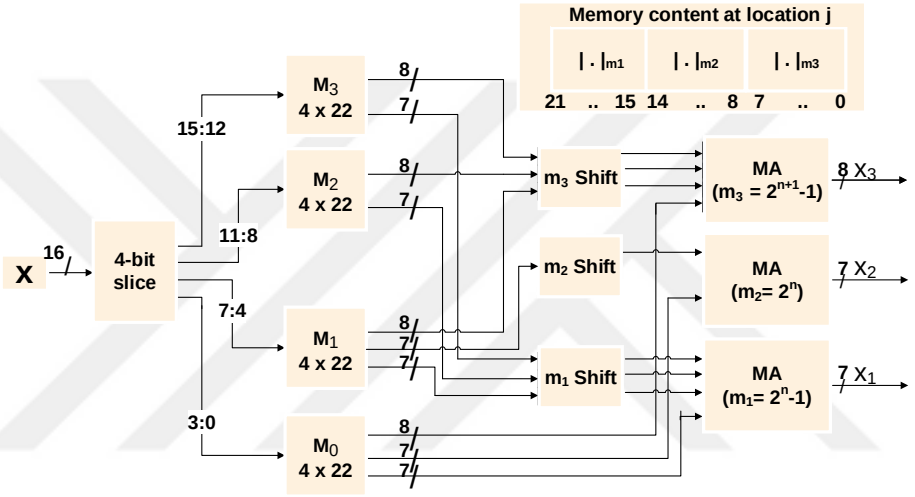
3.3.2 DWT implementation using RNS

The DWT implementation that employs RNS has mainly three components which are the forward converter, the modulo adders, and reverse converter. The forward converter, which is also known as the Binary-to-Residue Converter (BRC), is used to convert a binary input number to residue numbers. In contrast, the reverse converter or the Residue-to-Binary Converter (RBC) is used to obtain the result in a binary format from the residue numbers. We will refer to the RNS-system, which does not include RBC, as a forward-converter and modular-adders (FCMA), as illustrated in Figure 3.3.a.

The forward converter. The forward converter is used to convert the result of multiplying an input number by a wavelet coefficient to q residue numbers via LUT, shift and modulo adders, where q is the number of channels.



(A) The RNS Model



(B) The BRC block diagram

Figure 3.3 : The block diagram of DB2 RNS-based architecture. a) The complete model. b) The block diagram of the BRC for the 3-channel RNS-based, $P_7 = \{127, 128, 255\}$.

RNS-system number conversion. The received samples and wavelet coefficients span the real number and might take small values. One of the main drawbacks of RNS-number representation is that it only operates with positive integer numbers from$[0, M-1]$. The DWT coefficients are generally between 1 and -1. As a possible solution, we have divided the range of RNS, $[0, M-1]$, to handle those numbers.

In addition, the received sample $x[i]$ is scaled up by shifting y positions to the left (multiplying by 2^y), which ensures that $x[i]$ is a y-bit fixed point integer. In a similar manner, the wavelet coefficients are scaled by shifting it z positions to the left.

Modulo m_i multiplier. The multiplication of the received sample, $x[i]$, by the filter coefficients, which are constants, is performed by indexing the ROM. As the

word-length, w, of the received sample $X[i]$ is increased, the memory size becomes 2^w. In addition, q ROMs are required to perform the modulo multiplication.

We propose few improvements to this design. First, instead of preserving a dedicated memory for each modulo m_i, a ROM that contains all module results is used. Thus, each word at location j contains the q modules of $h_k.j.2^{11}$. Figure 3.3.b shows the internal BRC block design of the 3-channel moduli set $P_7 = \{127, 128, 255\}$ with its memory-map at right top corner. It shows that, for a location j, the least significant 8-bit contains $|h_k.x|_{m_3}$, the next 7-bit contains $|h_k.x|_{m_2}$ and the most significant $7-$bit contains $|h_k.x|_{m_1}$, which can be generalized as shown in equation (3.8) . The advantage of this method is that no extra hardware is required to separate each module value.

$$ROM(j) = |h_k.j.2^{11}|_{m_1}.2^{2n+1} + |h_k.j.2^{11}|_{m_2}.2^{n+1} + |h_k.j.2^{11}|_{m_3}, \quad j = [0, 2^w] \quad (3.8)$$

As with DAA-based approach, if the input word-length is 16 bits, the ROM should contain 2^{16} locations. One way to reduce the size of the memory is to divide it into four ROMs of 4×22. Figure 3.3.a shows the block diagram of the binary-to-residue converter with four ROMs; each is indexed by four bits of x. However, the output of each ROM should be combined, so that the final result can be corrected. It is worth noting that this division comes with a cost in terms of adders and registers.

According to the previous improvements, the RNS-based works are as follows. The input $X_{16-bit} = (x_1, x_2, x_3, x_4)$ is divided into four segments. Each of the $4-$bit segment is fed into one ROM, so that three outputs, corresponding to $|h_k.x_l.2^{11}|_{m_i}$, are produced. To obtain the final multiplication's result, each m_i output should be shifted by l positions, where l is the index of the lowest input bit (4, 8 or 12). The modular multiplication and shift for $2^n - 1$ and $2^{n+1} - 1$ can be achieved by a left circular shift (left rotate) for l positions, whereas the modular multiplication and shift for 2^n can be achieved by a left shift for l positions [55]. Finally, the modulo adder adds the corresponding output.

Modulo Adder (MA). The modulo adders are required for adding the results from a modular multiplier as well as for a reverse converter. In this work, we have two MAs—i.e., one is based on 2^n and the others are based on $2^n - 1$. Modulo 2^n adder is just the lowest n bits of adding two integer numbers, where the carry is ignored.

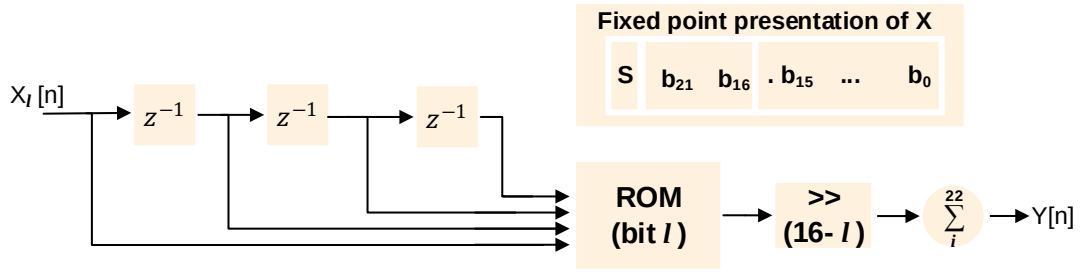


Figure 3.4 : The block diagram of $(2^n - 1)$ modulo adder.

Figure 3.4 shows the block diagram of the $2^n - 1$ modulo adder. It shows that MA is slow and less efficient than the conventional binary adder because MA needs one adder and one subtractor to perform the modulo addition. Therefore, the output of each MA will be delayed by one clock cycle compared to the conventional adder. **The Reverse Converter.** The Chinese remainder theorem (CRT) [53] provides the theoretical basis for converting a residue number into a natural integer. The moduli set $P_n = \{2^n - 1, 2^n, 2^{n+1} - 1\}$ can be efficiently implemented by four modulo adders and two multiplexers [57]. The output of the RBC is unsigned $(3\,.\,n\,+\,1)$-bits integer number. The actual signed number can be found by shifting the result $y+z$ positions to the left, which is equivalent to dividing by $2^{(y+z)}$. Both y and z are the scaled values of the input and wavelet coefficients, respectively. Generally, the word-length of 1-level DWT is bounded by equation (3.9) and should not exceed $(3.n-2)$ bits.

$$3n+1 \geq y+z+3 \tag{3.9}$$

3.3.3 Hardware complexity

Memory usage. DAA and RNS techniques employ the memory as a key resource to avoid multiplying two input variables. In each approach, as the number of filter taps increases, both the size and number of memories change. Assuming that the length of the received word is w-bit and there are N filter taps, the size of a memory element can be considered as $a \times b$, where a and b are the word-length in bits of the input and output, respectively. The value of determines the size of the memory, 2^a.

The total number of memory elements that are occupied by the DAA-based filter is $w.(N \times 22)$. The output is a fixed 16-bit fixed-point and the word-length is 22-bit. The number of memory elements remains constant as the filter taps increase, whereas the size of the memory exponentially increases to 2^N.

Table 3.1 : Occupied memories when DAA- and RNS-based approaches are used. The word-lengths, w, are $22-$bits and $16-$bits for DAA and RNS-based, respectively.

	DB2	DB4	DB10
Number of Filter taps	4	8	10
DAA Memory Usage	$22(4\times 22)$	$22(8\times 22)$	$22(10\times 22)$
RNS Memory Usage	$16(4\times 22)$	$32(4\times 22)$	$40(4\times 22)$

On the other hand, the total number of memory elements occupied by an RNS-based filter is $N.\lceil \log_2(w)\rceil.(4\times 22)$. This equation shows that the number of memory elements increases linearly with the number of filter taps, while the memory size remains constant (4×22). Table 3.1 shows a comparison of the memory usage with $w = 16$ for different DWT families.

Shift register and adder counts. DAA-based implementation employs shift registers and adders to sum the result at each bit level (Figure 3.2). For a word-length w with m magnitude bits, we need $(w-1)$ shift registers and $(w-1)$ 2-input adders (data combined by a tree adder architecture). To handle the negative numbers, the two's complement operation requires additional $(m-1)$ shift registers and $(m-1)$ adders. Thus, for l-level DAA-based implementation, a total of $l.(w-m-2)$ shift registers and 2-input adders is required.

On the other hand, for a word-length w and N-tap filter, the q-channel FCMA implementation requires N BRC blocks and $(q(N-1))$ MA blocks to compute the final result. Each BRC block has $(\lceil \log_2 w\rceil)-1$,$(\lceil \log_2 n\rceil-1)$ and $(\lceil \log_2 w\rceil)-1$ MA blocks for $2^n-1, 2^n$ and $2^{n+1}-1$ modulo, respectively. The modulo 2^n requires $log_2 n$ because shifting operations is not circular where shifting n-bit numbers to the left by n positions or more is always zero. Likewise, the RBC has 4 MA blocks (for $2^{n+1}-1$), 2 multiplexers and two subtractors. Thus, the total number of MA blocks at 1-level RNS-based is:

$$MA_t = 2N\ ((\lceil \log_2 w\rceil-1)_{2^n-1})\ +\ (\lceil \log_2 n\rceil-1)_{2^n} + q(N-1)+4 \tag{3.10}$$

For instance, 3-channel DB2 implementation requires 9 MA blocks to sum the result and P_7 RNS-based implementation has a total of 45 MA blocks when $w = 16$. Meanwhile, the number of RNS-based adders depends on the design of the MA block. For example, each MA block of (2^n-1) and $(2^{n+1}-1)$ requires 2 adders, while each MA block of 2^n requires 1 adder. Thus, $a_t = 12N+N(\lceil \log_2 n\rceil-1)\ +\ 5(N-1)+10$

Table 3.2 : Memory usage and adders for 1-level N-tap DAA- and RNS-based DWT.

	DAA-based	RNS-based
Memory usage	$w\ (N \times 22)$	$N\ \lceil \log_2 w \rceil (4 \times 22)$
Num. of adders	$w-m-2$	$17N+N(\lceil \log_2 n \rceil -1)+10$

adders are required, which can be simplified as follows (summarized in Table 3.2):

$$a_t = 17N + N(\lceil \log_2 n \rceil - 1) + 10 \tag{3.11}$$

3.4 Simulation Results, Performance Analysis and Validation

Hardware analysis was carried out by using a Xilinx System Generator for DSP, which is a high-level software tool that enables the use of MATLAB/Simulink environment to create and verify hardware designs for Xilinx FPGAs. The hardware-software co-simulation design was synthesized and implemented on ML605 Xilinx Virtex 6.

The implementation of RNS and DAA is compared with the multiplier-accumulate based DWT structure (MAC). We also consider the direct DWT implementation using an IP FIR Compiler 6.3 (FIR6.3) block, which provides a common interface to generate highly area-efficient and high-performance FIR filters. For RNS implementation, the moduli sets of$P_7 = \{127, 128, 255\}$ and $P_{10} = \{1023, 1024, 2047\}$ were used.

3.4.1 Resource utilization and system performance

Table 3.3 summarizes the resource use by RNS-based components— i.e., FCMA and reverse converter (RBC). The RBC consumes fewer resources and less power. However, the operating frequency is equal in all models and greater than the entire RNS-based filter.

Table 3.3 : The resource use and system performance of the RNS components— i.e., FCMA and reverse converter. FCMA involves the forward converters and modulo adders.

Resources	RNS-based ($n=7$)		RNS-based ($n=10$)	
	FCMA	RBC	FCMA	RBC
Slice Registers	656	157	883	190
Slice LUTs	591	138	854	180
RAMB18E1	16	0	16	0
Max. Freq. (MHz)	291.2	311.62	283.85	298.67

Table 3.4 : Resource use and system performance for the DWT implementation.

				RNS-based	
	FIR	MAC	DAA	P_7	P_{10}
Slice Registers	282	661	167	767	1089
Slice LUTs	128	520	71	721	1055
Occupied Slices	58	188	60	240	358
RAMB18E1	0	22	0	16	16
Max. Freq. (MHz)	296	230	472	259	261

Table 3.4 summarizes the resource consumption of each filter implementation. It shows that the MAC and IP FIR-based implementations have 4 multiplier units (DSP48E1s) with maximum frequencies of 296 and 472 MHz, respectively. In contrast, the proposed approaches are more complex than MAC. However; DAA- and RNS-based implementations has 22 and 16 memory blocks (BRAMs) used to store the pre-calculated wavelet coefficients. It also shows that the number of slice registers, slice LUTs and occupied slices of P_{10} RNS-based is greater than one of P_7 because the former has 31 output signals, while the later has 22 output signals. As a result, the number of flip-flops is increased and the number of resources is approximately increased by one third, while the maximum frequency in both designs is greater than 235 MHz.

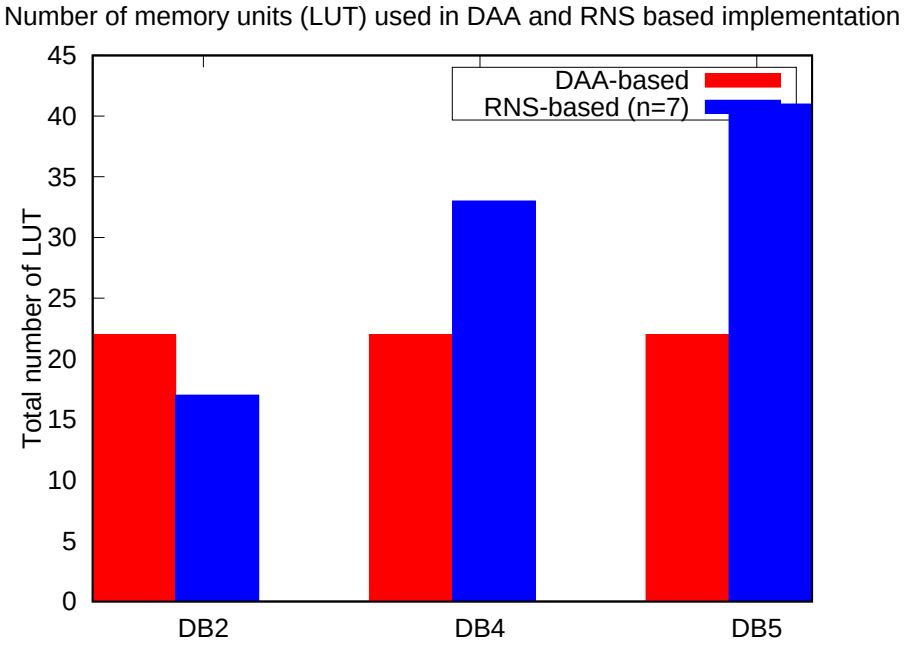


Figure 3.5 : Occupied memory by the first level of DAA and RNS-based DWT.

Table 3.5 : Resource use for the DWT implementation of DB2, DB4, and DB5.

Resources	DAA-based			RNS-based (P_7)		
	DB2	DB4	DB5	DB2	DB4	DB5
Number of Slice Registers	650	737	780	767	1441	1898
Number of Slice LUTs	521	539	568	721	1320	1677
Number of RAMB18E1	22	22	22	16	32	40
Max. operating frequency (MHz)	232.7	205.5	223.3	258.9	265.3	258.8

Table 3.6 : The SNR and PSNR values of 1-Level of different DWT implementations.

				RNS-based	
	FIR	MAC	DAA	P_7	P_{10}
SNR (dB)	83.2	78.7	70.4	53.41	54.78
PSNR (dB)	86.3	81.8	73.5	56.5	57.9

Table 3.5 shows a comparison between the DAA- and RNS-based 1-level DWT implementations when using larger filter banks— i.e., DB4 and DB5. It shows that DAA-based implementation occupies a fixed number of RAMB18E1. Figure 3.5 depicts the impact of using different wavelet families on memory. The number of memory elements of DAA-based implementation is fixed and depends on the word-length (Table 3.2).

However, as the number of filter taps increases, the memory size is exponentially increased to 2^N. In contrast, the number of memory elements that are used in RNS-based implementation is linearly increased as the number of filter taps is increased. Similarly, the number of memories that are used at multilevel DAA-based and RNS-based implementations with the l-level would be an aggregate of levels 1 through l.

Table 3.6 shows the performance based on the signal-to-noise-ratio (SNR) and peak-signal-to-noise-ratio (PSNR). Both DAA- and RNS-based approaches offer high signal quality with a peak signal-to-noise ratio (PSNR) of 73.5 and 56.5 dB, respectively.

Table 3.7 : Comparison of the current work with existing DWT architectures in literature.

	Jarrah [58]	Mamatha [59]	Madishetty [60]	Avinash [61]	Current Work	
Technology	Artix-7	Virtex-6	Virtex-6	Virtex-5	Virtex-6	
Wavelet Type	db1(Haar)	db4	db3	db2	db2	
Architecture	MAC	MAC *	AI **	DAA	DAA	RNS
Word-length	N/A	8-bit	8-bit	8-bit	22-bit	
Slice Reg.	2247 (1.77%)	139 (0.046%)	2890 (0.95%)	66 (0.52%)	661 (0.22%)	1089 (0.36%)
Slice LUTs	N/A	417 (0.39%)	5470 (5.17%)	136 (1.01%)	520 (0.49%)	1055 (0.99%)
Occupied Slices	N/A	N/A	N/A	61 (86%)	188 (0.5%)	358 (0.95%)
Freq. (MHz)[a]	214.6	633.4	344	301.8	229.8	261.0
Power (mW)[b]	101	632	204	N/A	66.54	53.05
DSP48Es	N/A	16	N/A	0	0	0
Throughput	N/A	1 ip/op	1 ip/op	8 ip/op	2 ip/op	2 ip/op

[a]Maximum Frequency.

[b]Dynamic Power.

*Parallel MAC Loop Based Filter (PMLBF); a pipelined real-time architecture.

**Algebraic Integer (AI).

3.4.2 Comparison

Table 3.7 provides a quantitative, comprehensive and a comparative study of different implementation method of DWT, including the convolutional MAC [58], Parallel MAC Loop Based Filter (PMLBF) [59] and Algebraic Integer (AI) [60] as well as our implementation of 1-D 1-level DWT. From the data, we observed that the proposed 22-bit DAA-based architecture almost has the same complexity as the lowest complexity implementation, as seen in [59]. However; our DAA- and RNS-based architectures consume less power. Even if the architectures in [59] and [60] have a throughput of one sample per clock cycle, the lower power required by DAA- and RNS-based implementations makes them superior in terms of throughput to power consumption.

3.4.3 Discussion

Hardware availability and system performance requirements are critical for selecting the appropriate architecture of the embedded platform. The number filter taps and word-length have a substantial influence on the overall.

Because the only difference between P_7 RNS-based and P_{10} implementations is the signal width, the maximum operating frequencies slightly changes. Furthermore, the 1-level DB2 filter bank was designed with maximum operating frequencies of 232 and 258 MHz for DAA- and RNS-based approaches, respectively. However, all high-frequency implementations introduce a latency of at least 10 clock-cycles for 1-level DAA-based DWT.

Another critical parameter that affects the DWT performance is the filter order. DAA-based implementation outperforms the RNS-based with at most 10-tap. When the number of taps increases, the number of memory units and binary adders within the RNS-based implementation constantly increases, while the memory-size is not affected (Table 3.2). The memory requirement for DAA-based implementation is exponentially increased as the number of filter taps increases.

3.5 Conclusion

The performance requirements and hardware resource availability has a great impact on picking up the proper algorithm for implementing a Discrete Wavelet Transform (DWT) on FPGA. In this paper, we addressed the effect of increasing the number of filter taps and word-length, which have a substantial influence on the overall performance of the design and resource availability. Moreover, we presented pipelined DAA- and RNS-based implementations and compared them with the pipelined MAC-based approach. DAA- and RNS-based approaches are multiplierless architectures that intensively use memory to speed up the entire processing time. The trade-off between system performances and resource consumption was also presented. Experiment results show that the DAA-based approach is appropriate for a small number of filter taps, while the RNS-based approach would be more appropriate

for a number of filter taps that are greater than 10, yet both DAA- and RNS-based approaches offer high signal quality with PSNR as 73.5 and 56.5 dB, respectively.

4. AN OPTIMIZED TWO-LEVEL DISCRETE WAVELET IMPLEMENTATION USING RESIDUE NUMBER SYSTEM[1]

4.1 Introduction

Discrete wavelet transform (DWT) [27, 41, 43, 60, 62] is a linear signal processing technique that transforms a time domain signal to "wavelet" domain [32]. DWT is usually implemented using the Finite Impulse Response (FIR) filter bank structures [45]. Figure 4.1 shows a convolution-based five-tap FIR filter with five multipliers, known as multiplier-accumulator (MAC) structure. In Figure 4.1, the multipliers are involved in multiplying an input $x[n]$ with filter coefficients, b_i. The direct implementation of the N-tap filter requires N multipliers, which are an expensive resource in field-programmable gate array (FPGA). With regard to this fact, it is known and clear that convolution-based method requires massive computations, takes more physical space and consumes more power [63], hence, lifting-based (LS) [48] has been developed and implemented to improve these limitation. In this work [42], we

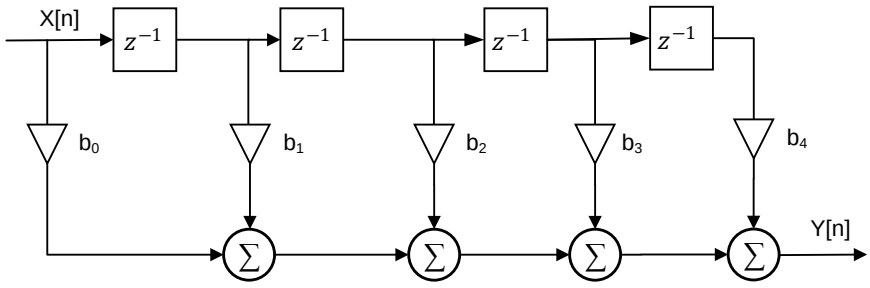


Figure 4.1 : 5-tap Finite Impulse Response Filter.

preferred the conventional convolution-based DWT implementation over the LS for the following reasons. In LS, as the critical path delay (CPD) increases, the energy per operation increases and the operating frequency decreases [64]. In [65, 66], the authors found out that as the length of the filter (N) increases, the CPD is increased, respectively. Hence, the sequence of multiplication and addition will be longer than

[1]This chapter is based on the paper "Alzaq, H.Y. and Ustundag, B.B. (2018). An optimized two-level discrete wavelet implementation using residue number system, EURASIP Journal on Advances in Signal Processing, 2018(1):41 "

the convolution-based scheme. Therefore, LS is observed to have poor scalability and is inappropriate for large filter lengths [59, 67]. In addition, LS requires temporary registers to store the intermediate results, which takes up more storage area and as well consumes more power [49, 68]. For these reasons, we decided to implement the DWT using the convolution-based approach, but with multiplierless architecture.

Multiplierless approaches eliminate the use of multipliers by replacing individual coefficient multipliers with a single multiplier block, known as a Multiple Constant Multiplications (MCM). Because filter coefficients are fixed and determined in advanced, the multiplication of filter coefficients by an input leads to area, delay, and power efficient architectures [69].

The existing multiplierless algorithms can be divided into two general classes; they either reduce the number of multipliers or totally replace them with a simplified circuit logic. The most popular reduction algorithms are Graph-based Eliminations (GE) [70] and common subexpression elimination (CSE) techniques [71–73]. The drawback of CSE algorithms is that its performance depends on the representations of the coefficients and also limited by the constant bit widths [72], whereas the GE require more computational resources due to a larger search space [74].

On the other hand, several multiplierless architectures that eliminate all multipliers have been proposed. Distributed arithmetic (DA) efficiently performs the inner product function in a bit-serial manner via a look-up table (LUT) scheme, followed by shift accumulation operations [39, 75, 76]. Based on our previous experience, we identified that the ROM size in DA-based structures increases with the increase in the word-length [42]. Residue Number System (RNS) is a highly parallel non-weighted arithmetic system that is based on the residue of division operation of integers using the look-up table (LUT) scheme [52, 77, 78]. The key advantage of RNS is gained by reducing an arithmetic operation to a set of concurrent, but simple, operations. Another advantage of RNS is its large dynamic range, which is divided into independent smaller ranges, where addition and multiplication operations are performed in parallel without a carry propagation among them. Several applications, such as digital filters, benefit from the RNS implementation, e.g. [54, 79, 80]. To the best of our knowledge, the aforementioned approaches consider only 1-Level DWT implementation.

Contribution of this paper. This article focuses exclusively on the implementation of 2-Level multiplier-free DWT. We propose a new design of 2-Level RNS-based DWT that efficiently uses the memory elements in the first-level DWT and do not employ any memory element in the next levels. In addition, this design eliminates the use of multiple residue-to-binary (RBC) converters between consecutive levels. Generally, the number of level is bounded by the output word-length and we determine it mathematically (equation 4.15 and 4.16). Finally, the proposed RNS-based approach could achieve high PSNR values with simple hardware structure and consume less power.

The remainder of this paper is organized as follows. In Section 4.2, the theoretical background on RNS is given. Section 4.3 illustrates the implementation of discrete wavelet transform. The implementation of the proposed 2-Level RNS is also presented. We further show an analytical comparison between these approaches. Section 4.4 presents the performance results. Finally, conclusions are drawn in Section 4.5.

4.2 Preliminaries

4.2.1 Discrete wavelet transform

The wavelet decomposition mainly depends on the orthonormal filter banks. Figure 4.2 shows a two-channel wavelet structure for decomposition, where $x[n]$ is the input signal, $g[n]$ is the high-pass filter, $h[n]$ is the low-pass filter, and $\downarrow 2$ is the down-sampling by a factor of two. By this way, each filter creates a series of coefficients that represent and compact the original signal information.

Mathematically, a signal $y[n]$ consists of high and low frequency components, as shown in equation (4.1). It shows that the obtained signal can be represented by using half the coefficients, because they are decimated by 2.

$$y[n] = y_{high}[n-1] + y_{low}[n-1] \tag{4.1}$$

The decimated low-pass filtered output is recursively passed through identical filter banks in order to add the dimension of varying resolution at every stage. Equations (4.2) and (4.3) mathematically express the filtering process of a signal through a digital high-pass filter $g[k]$, and low-pass filter $h[k]$. This operation corresponds to

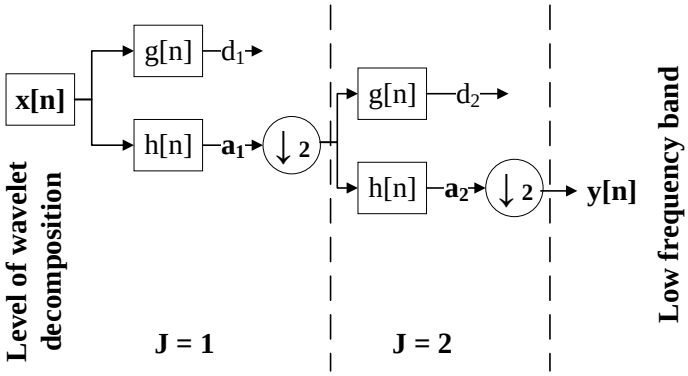


Figure 4.2 : Multi-resolution wavelet decomposition. The block diagram of the two-channel two-level discrete wavelet transform decomposition ($J = 2$) that decomposes a discrete signal into two parts. Note that $\downarrow 2$ is keeping one sample out of two, a_i and d_i are the approximation and details at level i, respectively.

a convolution with an impulse response of $k-$tap filters.

$$y_{high}[n] = \sum_{k} g[k].x[2n-k] \tag{4.2}$$

$$y_{low}[n] = \sum_{k} h[k].x[2n-k] \tag{4.3}$$

where n becomes $2n$, representing the down-sampling process. The output $y_{low}[n]$ provides an approximation signal, while $y_{high}[n]$ provides the detailed signal. There have been several wavelet filters proposed in literature, but in this paper, we have restricted ourselves to Daubechies wavelet filters only [33]. Because the down-sampling process follows each filter (as shown in Figure 4.2), equation (4.3) can be rewritten without the decimation factor as:

$$y[n] = \sum_{k=0}^{N-1} h[k].x[n-k] \tag{4.4}$$

where N is the number filter-tap. For the sake of simplicity of representing equation (4.4), $x[n-k]$ is replaced by $x[k]$.

4.2.2 Residue number system (RNS)

RNS [52, 77] is a non-weighted number system that performs parallel carry-free addition and multiplication arithmetic. In DSP applications, which require intensive computations, the carry-free propagation allows for concurrent computation in each residue channel.

Another aspect of using RNS is that an integer, within a large dynamic range, can be uniquely represented by set of residues,P, that are of much smaller values, corresponding to the size of the moduli set.

The RNS moduli set, $P = m_1, m_2, \ldots, m_q$, consists of q channels. Each m_i represents a positive relatively prime integer, that is GCD$(m_i, m_j) = 1$, for $i \neq j$.[2] Any number, $X \in \mathbb{Z}_M = 0, 1, \ldots M-1$, is uniquely represented in RNS by its residues $|X|_{m_i}$, which is the remainder of division X by m_i and M is defined in equation (4.5),

$$M = \Pi_{i=1}^{q} m_i = m_1 * m_2 * \cdots * m_q \tag{4.5}$$

M determines the range of unsigned numbers in $[0, M-1]$. In particular, M should be greater than the largest expected output.

In the RNS representation, addition and multiplication are performed entirely in parallel on each modulo,

$$Z = X \circ Y \xrightarrow{RNS} Z_{m_i} = |X_{m_i} \circ Y_{m_i}|_{m_i} \tag{4.6}$$

where $\circ$ represents the addition, subtraction or multiplication operation; and $m_i \in P$.

Mapping from the RNS system to integers, $\mathbb{Z}$, is performed by Chinese reminder theorem (CRT) [53, 57, 81]. The CRT states that binary/decimal representation of a number can be obtained from its RNS through equation (4.7), provided all elements of the moduli set are pairwise relatively prime.

$$|X|_M = (x_1, x_2, \ldots x_q) = |\sum_{i=1}^{q} \hat{M}_i |\alpha_i x_i|_{m_i}|_M \tag{4.7}$$

where $\hat{M}_i = M/m_i$ and $\alpha_i = |\hat{M}_i^{-1}|_{m_i}$ is the multiplicative inverse of $\hat{M}_i$ with respect to m_i.

The implementation of RNS-based DWT is obtained by substituting (4.6) in (4.4)

$$y[n]_{m_i} = y_{m_i} = |(\sum_{k=0}^{N-1} |h[k]_{m_i} . x[n-k]_{m_i}|_{m_i})|_{m_i} \tag{4.8}$$

for each $m_i \in P$. This implies that a $q-$channel DWT is implemented by q FIR filters that are working in parallel.

[2]The greatest common divisor (GCD) of two non-zero integers is the largest positive integer that divides them without a remainder.

For designing an efficient RNS-based DWT, the choice of the moduli set and hardware design of residue-to-binary conversion are two critical issues that should be considered. Most widely studied moduli sets are given as a power of two due to the attractive arithmetic properties of these modulo sets. For example, $\{2^n-1, 2^n, 2^{n+1}-1\}$ [55] $\{2^n-1, 2^n, 2^n+1\}$ [82] and $\{2^n, 2^{2n}-1, 2^{2n}+1\}$ [56] have been investigated. A 4-moduli set has been suggested to increase the dynamic range, e.g., $\{2^n-1, 2^n, 2^n+1, 2^{n+1}-1\}$ [83] and $\{2^n-1, 2^n, 2^n+1, 2^{2n}+1\}$ [84].

In this work, the moduli set $P_n = \{2^n-1, 2^n, 2^{n+1}-1\}$ is used for three reasons. First reason being that the modular adder is simple and identical for both $m_1 = 2^n-1$ and $m_3 = 2^{n+1}-1$. Secondly, for small $n = 7$, the dynamic range of P_7 is large and M is equal to 4145280, which would efficiently express real numbers in the range $[-2.5, 2.5]$ using 16-bit fixed-point representation, provided scaling and rounding are done properly. We assume that this interval is sufficient to map the input values, which does not exceeds ± 2. Thirdly, the reverse converter unit is simple and regular [57] because it does not employ any memory.

4.3 DWT Implementation Methodology

As mentioned in the previous sections, the wavelet transform of a signal can be performed by FIR filters, where the convolution operations are achieved by multiplying an input signal by the wavelet coefficients. In contrast, RNS-based approach has replaced the multiplication units with a suitable memory to perform the multiplication operations.

4.3.1 DWT implementation using RNS

Figure 4.3 shows the steps that are involved in RNS-based DWT approach. These steps are divided into offline and online steps. The offline steps are performed by converting the filter coefficients to RNS numbers and storing the result in a LUT. The online steps are used in converting each input values into RNS-system, before performing the filtering operations. Finally, the produced result is converted back to real-number system. We explained these process in section 4.3.1.1.

In General, the implementation of RNS-based DWT has essentially three components— i.e., the modulo adders (MA), forward and reverse converter. The

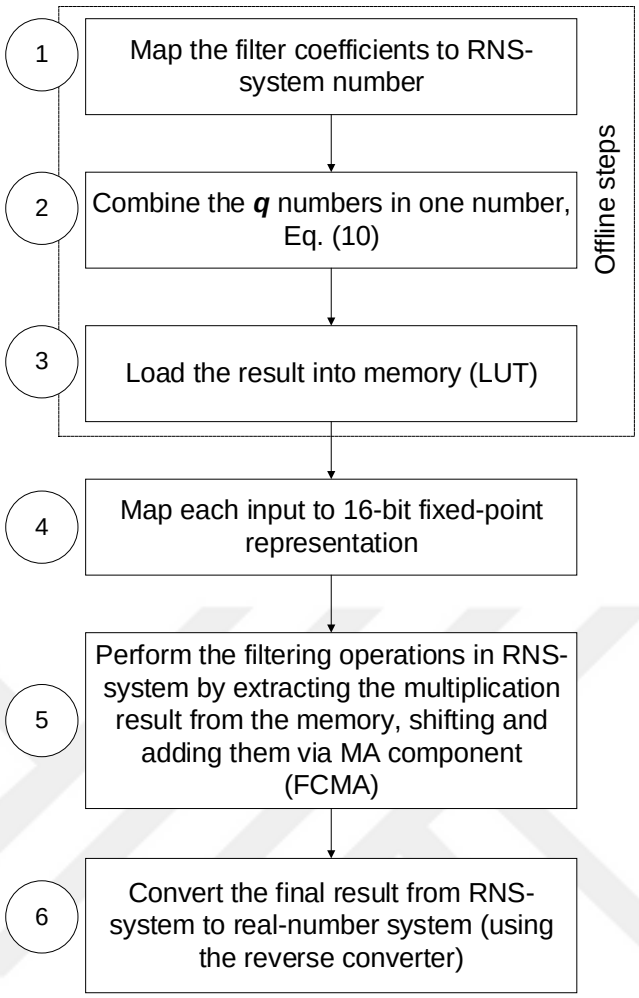


Figure 4.3 : The RNS-based DWT offline and online steps to perform the filtering operations. The offline steps are performed one time, while the online are performed for each input. q is the number of channels.

forward converter, also known as binary-to-residue converter (BRC), is used to convert a binary input number to residue numbers. In contrast, the reverse converter, also known as residue-to-binary converter (RBC), is used to obtain the result in a binary format from the residue numbers. These components are shown in Figure 4.4. We will refer to the RNS-system, which does not include RBC, as a forward-converter and modular-adders (FCMA), shown inside the dashed-line box in Figure 4.4.

4.3.1.1 Binary-to-Residue converter (BRC)

The BRC is used to convert the result of multiplying an input number by a wavelet coefficient to q residue numbers by using LUT, shift and modulo adders, where q is the number of channels. This procedure ensures that the multiplication operation is performed by using only memory.

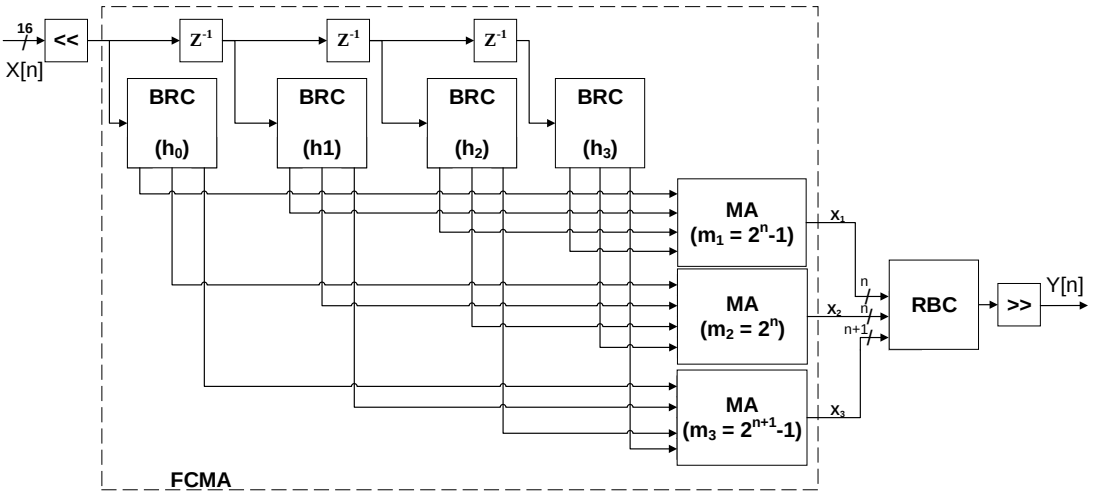


Figure 4.4 : The block diagram of DB2 RNS-based architecture. BRC stands for binary-to-residue converter, RBC stands for residue-to-binary converter and MA represents for modulo adder.

RNS-system number conversion: The received input and wavelet coefficients span the real number and might take small values. One of the main limitation of using RNS-number representation is that it only operates with positive integer numbers from $[0, M-1]$. The DWT coefficients are generally close to zero and between -1 and 1. Therefore, it is important to cope with both negative numbers and small numbers. To handle negative numbers, we mapped the real number to RNS range. Assuming the input samples are in $[-2.5, 2.5]$, we mapped any value in this range to a unique value in $[0, (M-1)]$. Any sample, which does not fit this interval, will produce incorrect values. Hence, the interval should be large enough to map all the numbers.

In principle, the received sample $X[i]$ is shifted y positions to the left (multiplying by 2^y, step 4 in Figure 4.3). This step ensures that $X[i]$ is a $y-$bit fixed point integer. In a similar manner, the wavelet coefficients are scaled by shifting it z positions to the left (step 1 in Figure 4.3). In our design, we set the filter scaling factor z to 11 and as a result, the coefficients of DB2 (equation 4.9) are multiplied by 2^{11} and rewritten as shown in equation 4.9.

$$y_{low}[n] = -266\,x[n] + 459\,x[n-1] + 1713\,x[n-2] + 989\,x[n-3] \qquad (4.9)$$

Modulo m_i multiplier: The multiplication of the received sample by the filter coefficients, which are constants, can be performed via indexing the LUT. It is critical to identify the size of LUT because as the word-length, w, of the received sample is

increased, the memory size becomes 2^w. Additionally, the design require q LUTs to perform the modulo multiplication.

We suggested several techniques to overcome these inadequate requirements. Instead of preserving a dedicated memory for each modulo m_i, one memory that contains all module results is used. In this scheme, each word at location j contains q modules of $h_k.j.2^{11}$. Figure 4.5 shows the internal BRC block design of the 3-channel moduli set $P_7 = \{127, 128, 255\}$ with its memory-map at the right top corner. This shows that, for a location j, the least significant 8-bit contains $|h_k.x|_{m_3}$, the next 7-bit contains $|h_k.x|_{m_2}$ and the most significant $7-$bit contains $|h_k.x|_{m_1}$, which can be generalized as shown in equation (4.10) (step 2 and 3 from Figure4.3). The advantage of this method is that no extra hardware is required to separate each module value. Table 4.1 shows the memory contents of h_1 in RNS-system using 4×22 LUT.

$$\begin{aligned} ROM(j) = &\mid h_k.j.2^{11}|_{m_1} \,.\, 2^{2n+1} \\ &+|h_k.j.2^{11}|_{m_2} \,.\, 2^{n+1} \\ &+|h_k \,.\, j.2^{11}|_{m_3}, \quad j = [0, 2^w] \end{aligned} \tag{4.10}$$

Table 4.1 : The memory content of $h_0 = -0.1294$ or $757(-266)$ multiplied by 2^{11} in $P_7 = \{127, 128, 255\}$ when word-length is 4 using eq. 4.10.

loc. i	$\mid -266.m_1\mid_{m_1}$	$\mid -266.m_2\mid_{m_2}$	$\mid -266.m_3\mid_{m_3}$	$ROM(i)$
0	0	0	0	0
1	115	118	244	3798772
2	103	108	233	3402985
3	91	98	222	3007198
4	79	88	211	2611411
5	67	78	200	2215624
6	55	68	189	1819837
7	43	58	178	1424050
8	31	48	167	1028263
9	19	38	156	632476
10	7	28	145	236689
11	122	18	134	4002438
12	110	8	123	3606651
13	98	126	112	3243632
14	86	116	101	2847845
15	74	106	90	2452058

It is obvious that if the input word-length is 16 bits, then the LUT-size becomes huge because 2^{16} locations will be needed. One way to reduce the size of memory is to divide it into smaller size, each consisting of $2 \times 22-$bits or $4 \times 22-$bits. Figure

4.5 shows the block diagram of the binary-to-residue converter with two and four memories, respectively. However, the output of each memory should be combined, so that the final result is correct. It is worth noting that this division comes with a cost in terms of additional adders and registers are used (discussed in section 4.3.4).

According to the previous improvements, the RNS-based system works as follows (step 5 from Figure 4.3). Suppose that four memories are used, each of 16 locations. The input $X_{16-bit} = (x_1, x_2, x_3, x_4)$ will be divided into four segments. Each $4-$bit segment will be fed into one memory, so that the 22-bit can be found, which will then be divided into three outputs, corresponding to $|h_k.x_l.2^{11}|_{m_i}$. We want to emphasize that this result is the multiplication of each 4-bit with a filter coefficient with respect to m_i.

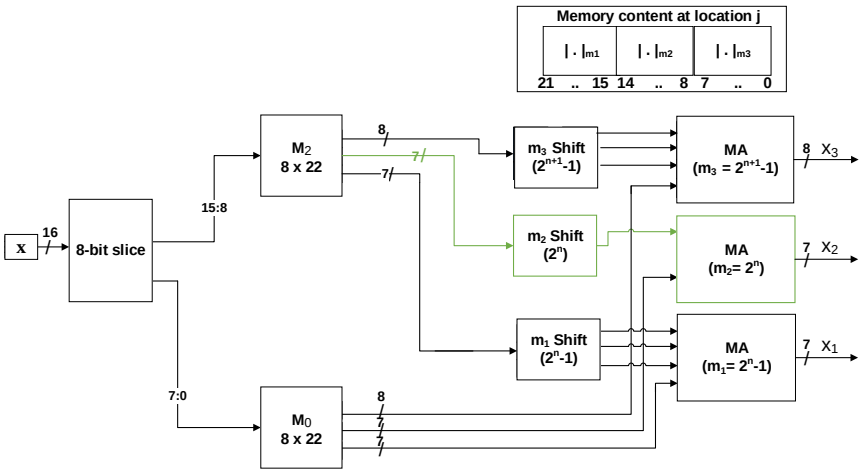
To obtain the final multiplication's result, each m_i output will be shifted by l positions, where l is the index of the lowest input bit (4, 8 or 12). The modular multiplication and shift for $2^n - 1$ and $2^{n+1} - 1$ can be achieved by a left circular shift (left rotate) for l positions, whereas the modular multiplication and shift for 2^n can be achieved by a left shift for l positions [55]. Finally, the modulo adder adds the corresponding output.

4.3.1.2 Modulo adder (MA)

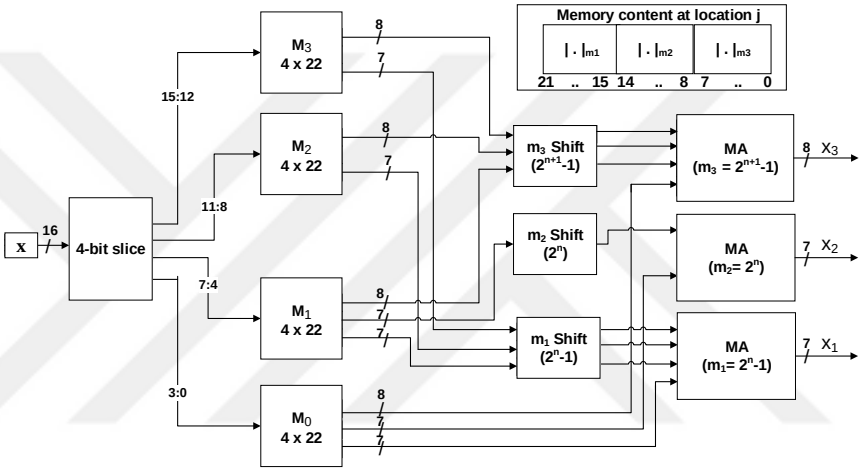
The modulo adders are required for adding the results from a modular multiplier as well as for the reverse converter. In this work, two types of MAs are necessary — i.e., the first one is based on 2^n and the other is based on $2^n - 1$. Modulo 2^n adder is just the lowest n bits of adding two integer numbers, where the carry is ignored. Modulo $2^n - 1$ adder differs from modulo 2^n adder in that the carry should be considered to limit the result to not be greater than $2^n - 1$, as in equation (4.11).

$$|x+y|_{2^n-1} = \begin{cases} x+y & \text{if } x+y \leqslant 2^n - 1, \\ x+y+1 & \text{otherwise} \end{cases} \tag{4.11}$$

To improve the design and enhance the speed, a parallel-prefix carry computational structure is used [85–87], which allows the implementation of highly efficient combinational and pipelined circuits for modular arithmetic.



(a) The BRC with two 8x22 memories



(b) The BRC with four 4x22 memories

Figure 4.5 : The block diagram of the binary-to-residue converter for the 3-channel RNS-based DWT, $P_7 = \{127, 128, 255\}$ (a) with two, (b) with four identical memories. The green blocks are not required because the shift result is always zero for m_2. The memory content at location j is as shown in the upper corner.

4.3.1.3 The reverse converter

The Chinese remainder theorem (CRT) [53] provides the theoretical basis for converting a residue number into a natural integer. The direct implementation of the CRT is inefficient because it requires a divider unit and several multipliers to determine the final output. However, the moduli set $P_n = \{2^n - 1, 2^n, 2^{n+1} - 1\}$ can be efficiently implemented by four modulo adders and two multiplexers (step 6 from Figure 4.3) [57]. Figure 4.6 shows the block diagram of the RBC of P_7, which is adapted from [57]. The output of the RBC is unsigned $(3n+1)$-bits integer number.

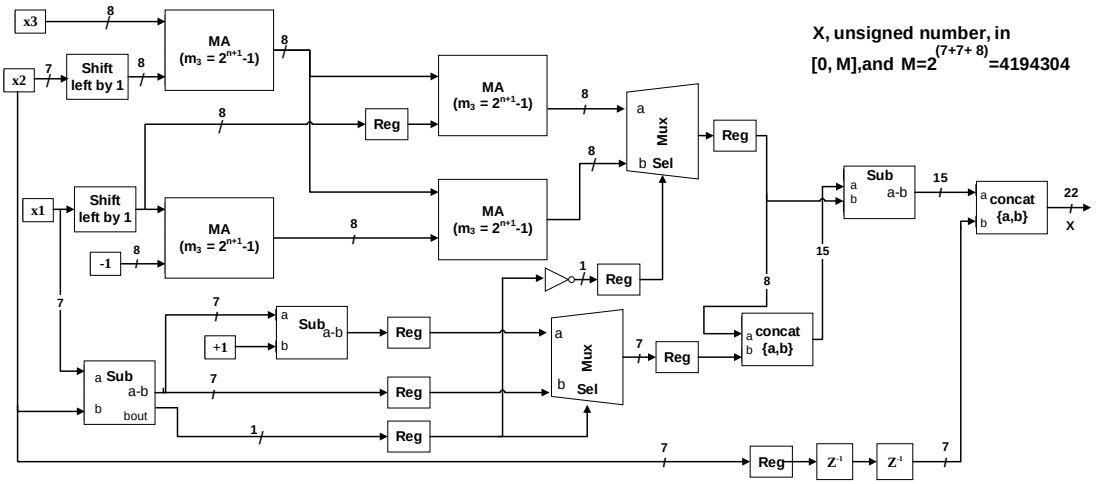


Figure 4.6 : The block diagram of the pipelined residue-to-binary converter for the 3-channel RNS-based DWT, $P_7 = \{127, 128, 255\}$. Four identical $2^{n+1}-1$ MAs, three subtractors and 2 multiplexers are generally required for RBC. Two shift registers are used for extending the 7-bit input by one bit. In the real implementation this operation does not add any cost because this operation is just equivalent to appending 0 at the first location of the input signal.

The actual signed number can be found by shifting the result $y+z$ positions to the left, which is equivalent to dividing by 2^{y+z}. y and z are the scaled values of the input and wavelet coefficients, respectively. Generally, the word-length of 1-Level DWT is bounded by eq. 4.12 and should not exceed $3n-2$ bits. Subtracting 3 is required because 2.5 is added to the input samples in order to have a 3-bit unsigned integer.

$$3n+1 \geq y+z+3 \tag{4.12}$$

4.3.2 Example

In this subsection, an example with input $x[1] = -0.4$ is given to illustrate how the RNS-based works. Each input is added to 2.5 and then multiplied by 2^8. Therefore, if $x[1] = -0.4$, the result is 538. Then this value is multiplied by the scaled $h_0 = -266$ in P_7. The sample input can be rewritten as $(0000\ 0010\ 0001\ 1010)_2$ or $x_1 = 0$, $x_2 = 2$, $x_3 = 1$ and $x_4 = 10$. These values are used to index the memory and the value of multiplying x_i by h_0 can be found in Table 4.1—i.e., 0, 3402985, 3798772 and 236689, respectively. From the table, the corresponding module of x_1 is $(0,0,0)_{P_7}$, x_2 is $(103,108,233)_{P_7}$, x_3 is $(115,118,244)_{P_7}$ and the corresponding module of x_4 is $(7,28,145)_{P_7}$. After that, the value of $h_0.x_2$ is

shifted 8-position corresponding to m_i and the value of $h_0.x_3$ is shifted 4-position corresponding to m_i. For the case of x_3, the mid value is shifted $8-$bit to the left, whereas the other values is circular shifted by $8-$bit to the left and the result becomes $(79,0,233)_{P_7}$. It is worth noting that the mid value is always 0 because the word-width is 7 and is less than shift value, 8. For the case of x_2, the mid value is shifted $4-$bit to the left, whereas the other values is circular shifted for 4 positions and the result becomes $(62,96,79)_{P_7}$. The sum of the partial results, performed via tree of 2-input MAs, is $(148,124,457)_{P_7}$ or $(21,124,202)_{P_7}$, which is equivalent to $(mod(-143108,127), mod(-143108,128), mod(-143108,255))$, respectively. Finally, the output of this memory-based multiplication is aggregated with next filter-taps using MAs.

4.3.3 2-Level DWT implementation

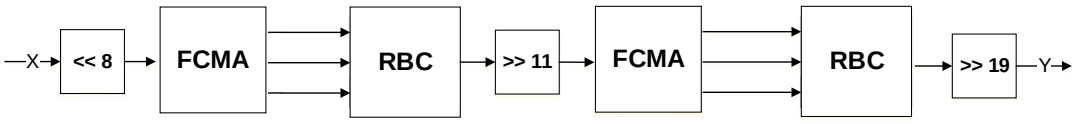
The 2-Level discrete wavelet transform comprises of two cascaded 1-Level DWTs (in series), where the output of the first level is fed into the second level (as shown in Figure 4.2). Figure 4.7.a shows the design of 2-Level RNS-based DWT, which involves two identical FCMA and two RBC blocks. The FCMA block is the RNS-based filtering (multiplication) block. It is obvious that converting between the number systems back and forth introduces some latency. Latency is defined as the number of clock cycles required to generate the first output sample once the input signal is applied. The latency, τ, of the 2-Level design is given by

$$\tau = 2(\tau_{FCMA} + \tau_{RBC}) \tag{4.13}$$

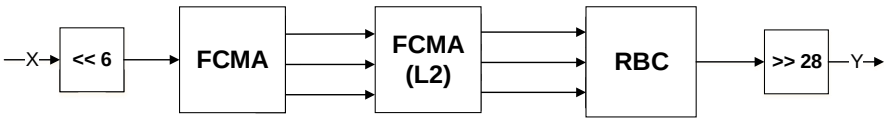
where τ_{FCMA} is the RNS-based filter latency and τ_{RBC} is the RNS-to-binary converting delay, respectively.

In this work, we suggest to eliminate the first RBC and feed the output of the first FCMA block into the second block, as shown in Figure 4.7.b. The advantage of this elimination is that the final output will be solely computed by one RBC component and one shift register. As a result, the latency becomes:

$$\tau = 2\tau_{FCMA} + \tau_{RBC} \tag{4.14}$$



(a) 2-level RNS-based design, (n=10)



(b) The proposed 2-level RNS-based Design, (n=10)

Figure 4.7 : The block diagram of 2-Level RNS-based DWT design (a) the full design with two RBCs, (b) shows the proposed design, which eliminates the middle RBC to improve the latency. FCMA represents FIR filtering process in RNS.

The only restriction is that the range of the used moduli-set should be greater than the maximum expected value, th_o, which can be computed as follows:

$$th_o = (\sum_k h_k)^2 \max(x[n]).(2^z)^2\ 2^y \leqslant M-1 \tag{4.15}$$

where h_k is the k^{th} DWT coefficient; $x[n]$ is the input; y and z are the input and filter scaling factors, respectively; and M is the maximum range from equation (4.5). As a consequence, the word-length of 2-Level DWT is bounded by eq. 4.16 and should not exceed $3n-2$ bits.

$$3n+1 \geq y+(2z)+3 \tag{4.16}$$

where $(3n+1)$ is the moduli-set word-length. Eventually, equation 4.16 can be generalized to any DWT level,(l), by using the following inequality:

$$3n-2 \geq y+(l.z) \tag{4.17}$$

4.3.3.1 The design of the second FCMA

The FCMA of the proposed stage can be implemented by two different techniques. The first one is based on memory that stores the multiplication result of multiplying an integer by the filter coefficient (Figure 4.8.a) and the second one is based on shift-add operations of the input by 2^r, where r is the position of each '1' of the filter coefficient (Figure 4.8.b). For example, from Table 4.1, h_0 is equal to 757 or $(01011110101)_2$, which is required to (circular) shift the input by $0,2,4,5,6,7,9$ and adding them to find the final multiplication result. The key feature of this technique is that each

multiplication operation is a left (circular) shift, which can be implemented by rewiring the order of the input bits. Indeed, the implementation is simple, and does not require any special circuits except several MAs to sum up the result. In the following sections, we refer to the optimized FCMA as memory-based, if it employs memory elements or shift-based, if it does not employ any memory.

4.3.4 Hardware Complexity

4.3.4.1 Memory usage

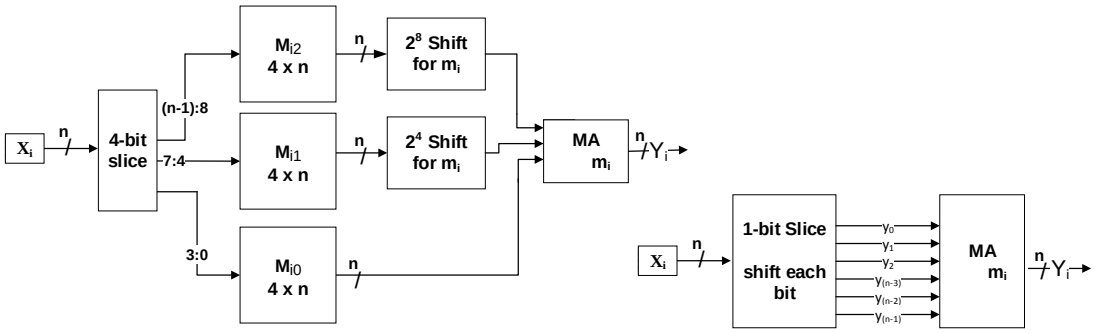
RNS techniques employs memory elements as a key resource to avoid multiplying two input variables. As the number of level increases, the number of memory elements changes. Assuming that the length of the received word is $w-$bit and there are N filter-tap, we define the size of a memory element by $a \times b$, where a and b are the word-size of input and output, respectively. The value of a determines the size of the memory, 2^a. The total number of memory elements that is occupied by an RNS-based filter is $N \lceil w/a \rceil$ of $(a \times b)$. This equation shows that the number of memory elements increases linearly with the number of filter taps (as shown in Figure 4.8.(a)), while the memory-size remains constant $(a \times b)$, but increased as a and b increase. The only overhead with large a is that the number of MA, which are required to sum up the final result, is increased. Table 4.2 shows a comparison of the memory usage when $w = 16$.

Table 4.2 : Occupied memories that are used by RNS-based DB2 DWT approaches. The input word-length, w, is $16-$bit and $b = 3n+1$.

Approach	Occupied memories	
	$a = 8$	$a = 4$
Memory-size	$(8 \times b)$	$(4 \times b)$
1-Level	8	16
2-Level	16	32
Optimized FCMA (memory-based)	24	36
Optimized FCMA (shift-based)	0	0

If the FCMA at the second stage is implemented by means of memory (memory-based FCMA), then the number of required memory can be calculated as follows:

$$qN\left(\lceil \frac{n}{a} \rceil\right) \tag{4.18}$$



(a) Memory-based FCMA (b) Shift-based FCMA

Figure 4.8 : The proposed FCMA block diagram of the second stage module, which directly performs a residue number multiplication by the filter coefficient at channel q. The M_{i0}, M_{i1} and M_{i2} blocks are identical memories of $4 \times n$ and contain values of $m_i.2^z.h_k.j$, where h_k is the k^{th} filter tap and $j \in [0,16]$. (a) Using memories. (b) Using shift operators and MAs.

For instance, a 3-channel of P_{10} FCMA has 6 memories of (4×10) and 3 memories of (4×11). Therefore, the FCMA at the second stage requires 36 memory elements and in total 52 are required for the whole design.

The design of FCMA at the second stage can be improved by eliminating all the memory at FCMA, as shown in Figure 4.8.(b). In this case, the proposed shift-based FCMA performs the multiplying operations via shift operations and MA units. The shift operation is always performed via rewiring the bits [55], which has no cost in terms of delay and hardware.

4.3.4.2 Adder counts

In addition to the memory complexity, we could derive an expression for the overall adder counts. In the following analysis, we can neglect the difference between $(2^n - 1)$ and $(2^{n+1} - 1)$ MAs because it will not affect the total number of MAs.

For a word-length w and $N-$tap filter, the $q-$channel FCMA implementation requires N BRC blocks and $(q(N-1))$ 2-input MA blocks to compute the final result (Figure 4.4). Each BRC requires at most $(q(w/a-1))$ 2-input MA blocks (Figure 4.5). Likewise, the RBC has 4 MA blocks, 3 subtractors, and 2 multiplexers. Thus, the total number of MA blocks at 1-Level RNS-based is given by:

$$\begin{aligned} MA_t &= qN((w/a)-1) + q(N-1) + 4 \\ &= qN\,(w/a) - qN + qN - q + 4 \\ &= q(N\,(w/a) - 1) + 4 \end{aligned} \tag{4.19}$$

For instance, 3-channel DB2 implementation requires 9 MA blocks to sum up the final result, and in total P_7 RNS-based implementation has a total of 49 MA blocks when $w = 16$ and $a = 4$ bits.

If the FCMA of the second stage is implemented by means of memory, then each unit requires $(q.((\lceil \frac{n}{a} \rceil) - 1))$ MAs to sum the output of each tap (Figure 4.8.(a)). In addition, $(q(N-1))$ MAs are required to sum all the output of all taps. This means, $(N\ q((\lceil \frac{n}{a} \rceil) - 1) + q(N-1))$ MAs are used for the memory based proposed approach.

In contrast, if the FCMA of the second stage is implemented by rewiring the input of the first stage, then each tap requires at most $q(n-1)$ MAs, where n is the channel width. However, not all of these MAs are required, because the shift operations are applied to the binary ones of the filter coefficients. For example, from Table 4.1, h_0 equal to $(757)_{10}$ or $(01011110101)_2$. This means that the proposed approach requires $q(\bar{n}-1) = 3.6 = 18$ modulo adders, where $\bar{n}$ is the average of binary ones in $h_j.2^z$. Table 4.3 summarizes the number of memories and MAs of each implementation, respectively.

It is clear that as (w/a) increases, the number of MA increases because $(w/a-1)$ MA are required to construct MA tree. Hence, the critical path delay (PSD) involves one multiplier followed by $log_2((w/a)-1)$ levels MA tree. As a consequence, there is a trade-off between the number of memory and its size on the overall performance of the system.

Table 4.3 : Memory usage and adders for RNS-based approaches for N-tap DWT.

	Number of Memories	Number of MAs
1-Level	$N\ (w/a)$	$q(N\ (w/a)-1)+4$
2-Level	$2N\ (w/a)$	$2(q(N\ (w/a)-1)+4)$
Memory-based FCMA	$qN\ (\lceil \frac{n}{a} \rceil)$	$qN\ ((\lceil \frac{n}{a} \rceil)-1)+q(N-1)$
Shift-based FCMA	0	$qN\ (\bar{n}-1)+q(N-1)$

4.4 Simulation Results, Performance Analysis and Validation

In the previous section, we have demonstrated the design of the DWT by using a residue number system. The 2-Level DWT RNS-based has been designed, implemented and tested with series of simulations to verify the DWT functionality. Experiments were carried out on the Xilinx ZC706 evaluation board [88]. The

performance of the proposed approach was compared with the distributed arithmetic (DA) [41], which is a multiplierless DWT. We also considered the direct DWT implementation using an IP FIR Compiler 6.3 (FIR6.3) block, which provides a common interface to generate highly parameterizable, area-efficient, high-performance FIR filters [89].

In the following experiments, the moduli sets of $P_7 = \{127, 128, 255\}$, $P_{10} = \{1023, 1024, 2047\}$ and $P_{13} = \{8191, 8192, 16383\}$ were used. The dynamic range of these sets are $M = 4161536$, 2144338944 and 1099310309376, respectively. In fact, the moduli sets of P_{10} and P_{13} are selected because their dynamic range are greater than th_o. For instance, equation 4.15 shows that $th_o = 1279020283$ for P_{10} with $y = 6, z = 11$ and $\sum(h_i) = 1.5436$. In all RNS-based implementations, the input word-length was set to 16 bits.

4.4.1 Resource utilization and system performance

Table 4.4 summarizes the resource use by RNS-based components— i.e., FCMA and reverse-binary converter (RBC) when using $2 \times b$ memory, where 'b' is the output word-length of P_n and equal to $(3n+1)$. The $2-$level, $3-$channel RNS contains two FCMA units and 3 different MAs (Figure 4.5.b). The RBC unit consumes fewer resources and less power with the operating frequency in all models being approximately equal and as high as 350 MHZ or more. The optimized shift-based FCMA consume less power but the number of occupied slices is doubled compared to the memory-based implementation. Finally, it is clear that the BRAM consumes most power in all memory-based models (approximately 16 mW).

4.4.2 2-Level DWT evaluation

Table 4.5 lists the resource consumption and the system performance for the 2-Level DWT implementations when two memory elements are used in each filter-tap. The FIR-based model shows better performance compared to all multiplierless architectures but it requires several multiplier units, known as DSP48E in modern FPGA [88].

It is also observed that the maximum frequency of all RNS-based schemes is higher than DA-based DWT. Because the only change amongst P_7, P_{10} and P_{13}

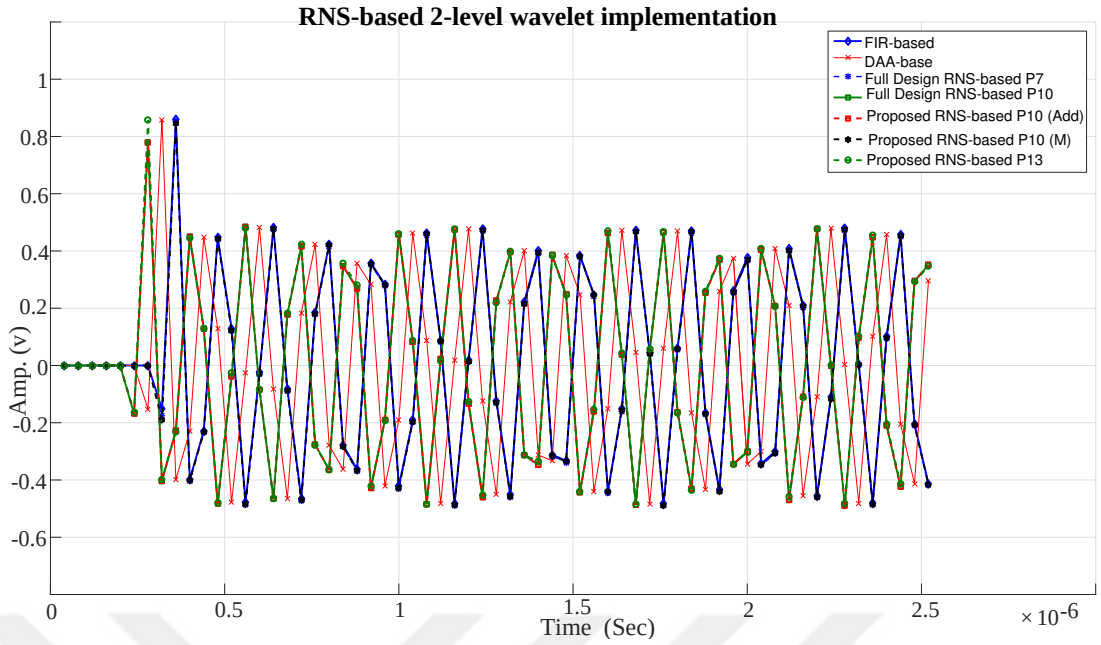


Figure 4.9 : Comparison amongst all 2-Level DWT approaches when a sinusoidal signal is applied.

implementations is the moduli-set width, the maximum operating frequencies slightly changes amongst these designs. Furthermore, the 2-Level DB2 filter bank was designed with maximum operating frequencies between 260 and 360 MHz for full and optimized shift-based FCMA, respectively. However, P_7 RNS-based is the only model that has less resources compared to DA-based because of its small word-length.

Furthermore, Table 4.5 shows that the RNS schemes consume less power compared to DA-based DWT. The exception is that the Full model of P_{13}, which occurred due to the hardware usage and the memory-size—i.e., 8×40. The large size of the memory can further be split into smaller memory (as shown in Figure 4.5). Table 4.6 presents the impact of using four-memory in each filter-tap on the power. It is obvious that the power consumption is increased as the number of memory and its size increase. The proposed shift-based RNS model compared to these models showed a better performance regards to power and maximum operating frequency. In spite of the deficiency in terms occupied slices, the proposed model has less number of memories and multipliers compared to DA- and FIR-based schemes.

Table 4.4 : FPGA resource utilization and system performance for the RNS components— i.e., FCMA and reverse converter. The FCMA involves the forward converters and modulo adders. "S-FCMA" is the shift-based FCMA, the one that rewires the input, and "M-FCMA" is the memory-based FCMA.

Resources	$(n=7)$		$(n=10)$		$n=10$	$n=13$	$n=10$
	FCMA	RBC	FCMA	RBC	S-FCMA[c]		M-FCMA
Number of Slice LUTs	234	114	335	143	731	999	348
Number of Slice Registers	375	148	478	187	792	1024	471
Number of occupied Slices	121	55	158	57	360	524	164
Number of RAMB18E1	8	0	8	0	0	0	24
Output word-length (bits)	22	0	31	0	31	40	31
Worst negative slack (ns)	7.3	7.2	7.1	7.29	7.23	7.26	7.29
Max. operating freq (MHz)	367.1	353.7	346.6	369.7	360.1	365.7	368.9
Data path delay (ns)	2.599	2.65	2.66	2.5	2.76	2.7	2.65
Estimated Power (mW)[a]	25	3	29	3	6	7	21
Block RAM Power (mW)	16	0	16	0	0	0	16
Latency (CC)[b]	5	6	5	5	6	6	5

[a]The IO power estimation is not considered.

[b]Clock cycle.

[c]Rewiring the input for implementing shift operations and a series of MAs.

Table 4.5 : FPGA resource utilization and system performance of 2-Level DB2 DWT implementation with ZC706.
"S-FCMA" is the shift-based FCMA, the one that rewires the input, and
"M-FCMA" is the memory-based FCMA.

Resources	FIR	DA	RNS-based ($n = 7$)	RNS-based ($n = 10$)			RNS-based ($n = 13$)	
			Full	Full	M-FCMA	S-FCMA	Full	S-FCMA
Number of Slice LUTs	92	1108	730	1000	882	1759	1261	2455
Number of Slice Registers	494	1250	1007	1307	1231	1648	1643	2204
Number of occupied Slices	122	411	351	434	444	656	561	885
Number of memory	0	44	16	16	32	8	16	8
Number of DSP	7	0	0	0	0	0	0	0
Worst neg. slack (ns)	7.654	6.01	6.59	6.87	4.73	6.6	6.86	6.2
Max. operating freq (MHz)	426.3	250.6	293.3	316	189.7	294.4	318.8	263.2
Data path delay (ns)	2.017	3.73	3.152	2.94	4.84	3.07	2.9	3.54
Estimated Power (mW)	13	63	42	50	55	44	81	63
Block RAM Power (mW)	0	37	20	23	29	8	46	15

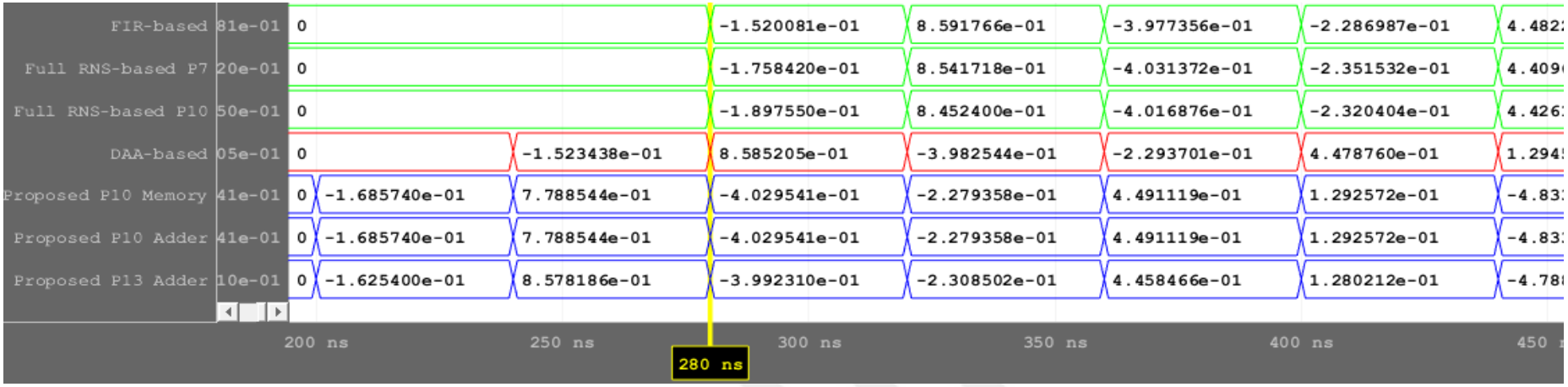


Figure 4.10 : The output of 2-Level DWT using DSP Logic Analyzer when a sin wave is applied. Each clock cycle is 10 ns.

Table 4.6 : The effect of using 4 memories in each filter-tap with ZC706.
"S-FCMA" is the shift-based FCMA, the one that rewires
the input, and "M-FCMA" is the memory-based FCMA.

Resources	DA	RNS-based ($n = 7$)	RNS-based ($n = 10$)			RNS-based ($n = 13$)	
		Full	Full	M-FCMA	S-FCMA	Full	S-FCMA
Number of Slice LUTs	92	963	1256	1739	1438	2119	2216
Number of Slice Registers	494	1240	1469	2047	1891	2048	2626
Number of occupied Slices	122	329	574	726	697	800	973
Number of memory	44	32	32	52	16	32	16
Output word-length (bits)	22	22	31	31	31	40	40
Number of DSP	7	0	0	0	0	0	0
Worst neg. slack (ns)	7.654	6.585	5.419	5.72	4.95	6.5	4.9
Max. operating freq (MHz)	426.3	292.8	218.3	233.5	198	285.7	196.1
Data path delay (ns)	2.017	3.369	4.21	3.99	4.357	3.316	4.7
Estimated Power (mW)	78	70	85	88	60	142	92
Block RAM Power (mW)	51	39	43	49	18	88	36

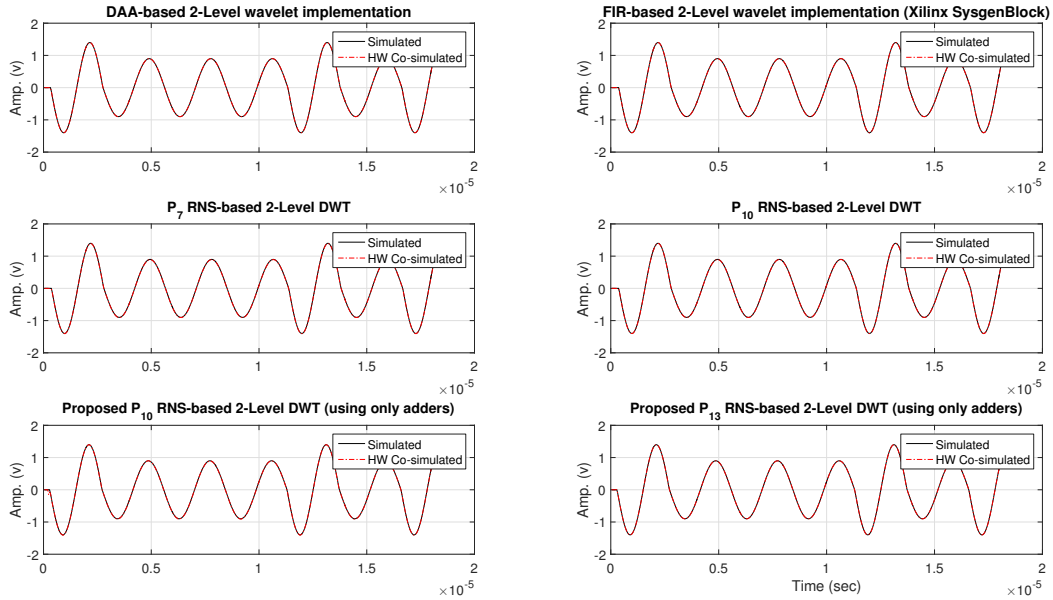


Figure 4.11 : The output of the 2-Level DWT when a pattern-based signal is applied [27]. It is used to extract the main features of the received signal.

4.4.3 Functionality verification

In this experiment, a sinusoidal signal is applied on each approach to verify the functionality of each design. Figure 4.9 shows a comparison amongst all RNS-based implementations with FIR- and DA-based implementations. It indicates that the proposed approaches are ahead of other implementations. To be more accurate, the 2-Level DB2 DWT implementations were simulated by DSP Logic Analyzer and the result is shown in Figure 4.10. It depicts that the signals resulting from Full RNS- and FIR-based lag behind the proposed architecture by 80 ns or 8 clock cycles, whereas the signal resulting from DA-based lags by 4 clock cycles.

Eventually, we have verified the simulated result on ZC706 development kit and the simulation and hardware co-simulation results of the 2-Level DB2 implementations are highly correlated, as shown in Figure 4.11.

4.4.4 Precision analysis

Generally, convolution-based DWT involves floating-point operations, which introduces rounding errors. Because the filter-banks coefficients, designing by means of floating-point, require large hardware resources to retain the precision, we replaced

the floating-point method with RNS numbering system. We simply multiplied the input by 2^y and the filter coefficients by 2^z. At the end, we converted the result back to floating-point number. PSNR is the most commonly used method to measure the quality of the result. In fact, it measures the peak error and high PSNR means better quality and that less error is introduced to the result.

We carried out the precision analysis for the first and second levels. The Daubechies wavelet with 4 coefficients were used and the result of each level was compared with the actual double-precision values via MATLAB. Table 4.7 shows the behavior of the proposed approaches in terms of input and wavelet coefficients precision. The output precision is set to $Q_{5,16}$ for all implementations. It is worth noting that employing the memory-based FCMA or shift-based FCMA at the second level has no effect on the output because both schemes compute identical results.

The optimized 2-Level with P_{10} has a maximum input scaling factor of 6(due to equation 4.16). As a consequence, we can not adapt their scaling factors. In contrast, the optimized 2-Level of P_{13} has higher input and filter coefficients scaling factors due to its large word-length, which enables it to have large accuracy values. The reason for preferring this 22-bit precision is that reducing the signal precision introduces small errors which propagate to the next level.

Table 4.7 : The PSNR values of 1- and 2-Level of different DWT implementations.

	DA	FIR	RNS			Optimized RNS	
			P_7	P_{10}	P_{13}	P_{10} [a]	P_{13} [a]
Architecture	1L/2L	1L/2L	1L/2L	1L/2L	1L/2L	2L	2L
Input Precision	$Q_{5,16}$	$Q_{5,16}$	$y = 8/8$ [b]	$y = 12/12$	$y = 13/12$	$y = 6$	$y = 11$
Coeff. Precision	$Q_{1,15}$[c]	$Q_{0,15}$	$z = 11$	$z = 16/11$	$z = 18/13$	$z = 11$	$z = 13$
Internal word-length	22-bit	NA	22-bit	31-bit	40-bit	31-bit	40-bit
PSNR (dB)	73.5/63.5	86.3/78.7	56.5/41.87	84/53	90/54	48.5[d]	54.5

[a]Optimized FCMA model, where one RBC is used

[b]The input to RNS circuit is 16-bit unsigned integer

[c]Memory word-length

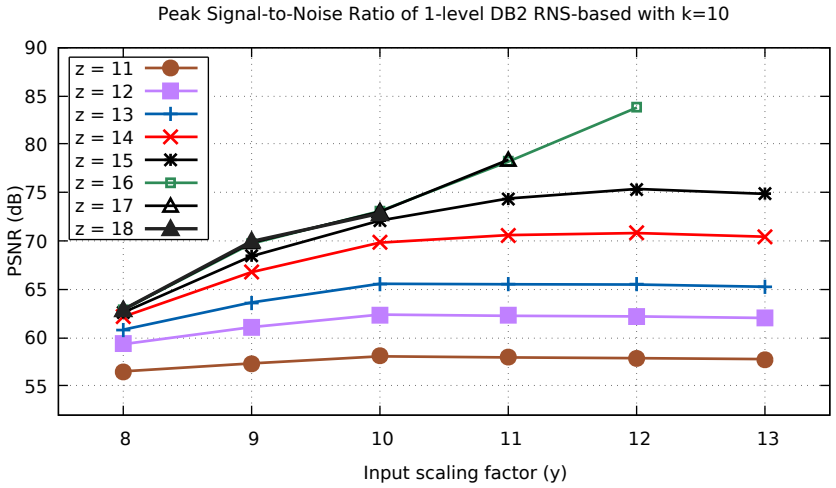
[d]This is the only combination that satisfies equation (4.16)

Table 4.8 : A Comparison between RNS-based implementations using moduli-set P_n.

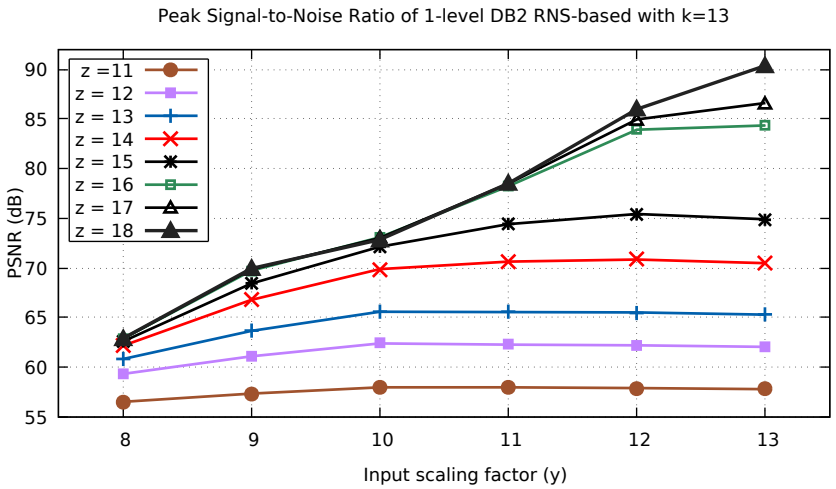
	P_7	P_{10}	P_{13}
Pros	• Design is simple • Consume less power	• Design is moderate • The shift-based scheme has lower latency and consume less power	• Can be extended to 3-Level (with $y = 8$ and $z = 9$) • The shift-based scheme has lower latency and consume less power
Cons	• Low PSNR • Can not be extended to multi-level DWT	• Can not be extended to 3-Level DWT	• Word-length is high • consume large resources

Table 4.7 presents that the maximum achieved PSNR of P_7 set is 56.5 dB and 41.87 for the first and second level, respectively. We could not achieve better accuracy with the specified scaling factors because $y+z+3 = 21 \leq (3*7)+1 = 22$ (see equation 4.16). If an application requires higher accuracy values, then different moduli set with large n should be selected. Figure 4.12 compares the effect of changing the scaling factors of two different moduli sets, P_{10} and P_{13} for 1-Level DB2 RNS-based approach. The input scaling factor was varied between 8-bit and 13-bit, and filter scaling factor was varied between 11 and 18. As expected, lower scaler factors produces PSNR equal to 60 dB. While the maximum PSNR equal to 90.37 is obtained with $y = 13$ and $z = 18$ for P_{13}, the maximum PSNR of P_{10} is obtained when $y = 12$ and $z = 16$ (equation (4.12)). However, both models have approximately identical PSNR values (limited by eq. 4.16), which implies that increasing the word-length from 31-bit (in P_{10}) to 40-bit (in P_{13}) has no effect on the accuracy. It is worth mentioning that as the filter scaling factor increases, no hardware cost is added to the design, because all the changes correspondingly occurred in the memory contents. Figure 4.13 shows the effect of the scaling factors for 2-Level DB2 RNS-based approach. The proposed 2-stage RNS-based P_{13} with $z = 13$, has maximum PSNR of 54.5 dB, which is roughly equal to the full model with $z = 11$. In addition, using higher moduli set has an effect on precision values (PSNR). Although the proposed 2-Level architecture achieves an

acceptable PSNR, 50 dB in the case of P_{13}, it should be noted that it is less than the achieved values by DA- and FIR-based.



(a) One-level PSNR when P_{10} is used.



(b) One-level PSNR when P_{13} is used.

Figure 4.12 : The impact of input, y, and wavelet filter coefficients, z, scaling factors of 1-Level RNS-based implementation on PSNR, with respect to a) P_{10} and b) P_{13} moduli sets.

4.5 Conclusion

In this article, we have addressed the development of a multiplierless scheme for 2-Level RNS-based DWT, which can be adapted to any moduli set, with any number of channel. This approach intensively use memory to speed up the entire processing time. In order to achieve low latency, we incorporated two novel ideas into the 2-Level

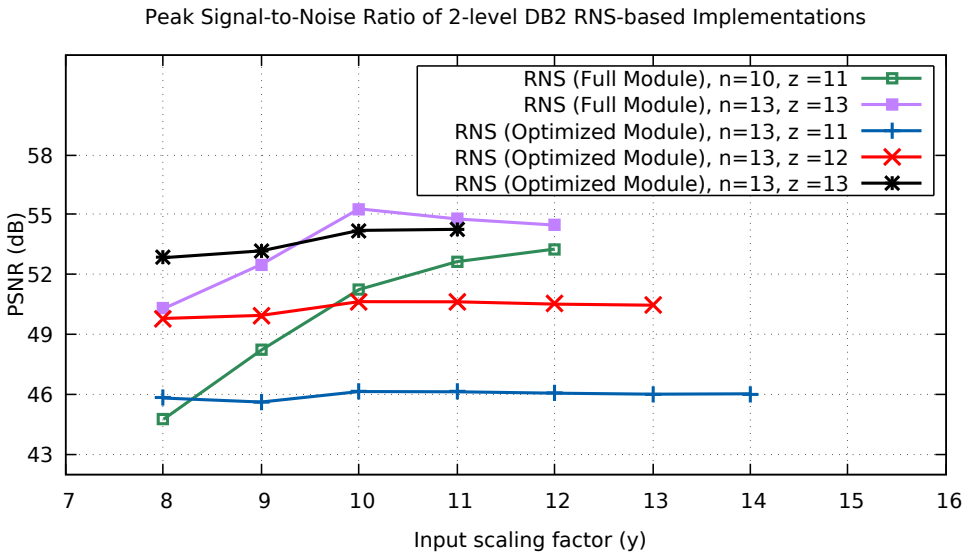


Figure 4.13 : The impact of input, y, and wavelet filter coefficients, z, scaling factors of 2-Level RNS-based implementation with respect to P_{10} and P_{13} moduli sets on PSNR.

proposed design, as follows: 1) eliminating the intermediate RBC unit; 2) replacing the internal memory of the second level by simple circular shift operations. A key feature of this approach is that the user can change the scaling factors, y and z, either to achieve high PSNR values or lowering the PSNR value in order to design multi-Level DWT with low latency.

The trade-off between system performances and resource consumption was addressed. Experiment results showed that the RNS-based approach would be more appropriate for multi-level DWT because the number of memory element is always constant as the number of level is increased. In addition, it is observed that the proposed RNS-based DWT implementation has lower latency than FIR and P_n RNS-based implementations. Finally, an acceptable precision can be achieved by adapting the scaling factors. Table 4.8 indicates the advantage and disadvantages of each moduli-set.

Given the implementation examples for experimental verifications and analysis, the approach was validated on a ZYNQ ZC706 development kit. The co-simulation results have also been verified and compared with the simulation environment. The complexity and optimization of multi-level DWT with respect to hardware structure provides a foundation for employing an appropriate algorithm for high-performance applications, such as in cognitive communication, where DWT analysis is combined with machine learning algorithms.

5. CROSS-LAYER ADAPTIVE MODULATION AND CODING FOR SPECTRAL EFFICIENCY COGNITIVE COMMUNICATION[1]

Abstract

Cognitive radio (CR) aims at improving spectral efficiency through dynamic management of the available spectrum. In contrast to the widespread approach in CR, a pattern-based cognitive communication system (PBCCS) was suggested to optimize non-periodic RF waveforms to maximize the link spectral efficiency (LSE) under low-SNR. PBCCS is a cross-layer approach that merges the channel encoding and modulation. The transmitter encodes sequences of bits into continuous signal patterns by selecting the proper symbol glossaries. In this article, we propose an architecture for pattern-based cognitive communication (PBCCS) implemented in embedded systems. We introduce an algorithm, called advanced sinusoidal pattern envelope construction (ASPEC), which improve the recognition rate of the received symbols. As a proof-of-concept, we develop, and experimentally validate a complete integrated software/hardware platform. The digital platform is mainly comprised of a Xilinx ZC706 board with an integrated transceiver chip ADI AD9361 RF, a high-performance ADC. Our solution reveals that the application of AMC based on glossary coding can achieve considerable spectral efficiency gain in PBCCS.

5.1 Introduction

Cognitive radios (CR) [3, 4] is defined as an intelligent radio systems that can be programmed and configured dynamically based on interaction with its environment. Currently, software defined radios (SDR), the key enabling technology of CR, combines the primitive communication system functions such as working frequency band, modulation and demodulation, channel access methods ... etc, with a software radio for real-time communication. SDR allows flexible configuration and dynamic

[1]This chapter is based on the paper "Cross-Layer Adaptive Modulation and Coding for Spectral Efficiency Cognitive Communication", which was submitted to IEEE Transactions on Cognitive Communications and Networking. Current status: Under Review

programming of the radio equipment through incorporation of artificial intelligence (AI) techniques into hardware. The fundamental functions of CR include (1) the ability to sense the environment (cognitive capability), (2) learns from the sensed information, (3) makes decisions (decision capability) and (4) the ability to adapt/configure its operating parameters (reconfigurable capabilities) in real-time [3, 4, 90].

However, while incorporating learning and adaptiveness into CRs is highly desirable, the cognitive principle will not be realized until the network layer, seamlessly incorporate intelligence [91]. Generally speaking, the CR functions span the top-down network architecture and require interacting of multiple layers with each other. Yet, this interaction does not fit top-down network architecture because it follows strict layering principles. To mitigate this problem, a cross-layer model was suggested to allow interaction between many layers by merging the functionality of adjacent layers together [92]. As an example, information about the available channels and its states at the physical layer can be be integrated with the automatic repeat request (ARQ) at the data-link layer to maximize the physical throughput [93].

This article presents a spectrum aware cross-layer protocol for pattern-based cognitive communication system (PBCCS) [27, 42]. PBCCS incorporates data encoding in adaptive modulation technique using special kind of signals for each digital data. The initial form of this work was published in [10,42], where the concentration is mostly at the receiver design. However, a comprehensive form of the whole idea, including the design of the glossary, is presented in this article. Basically, we consider a cross-layer design approach to enhance the recognition rate at the receiver by generating special kinds of symbols at the transmitter. The generated symbol, which composed of different patterns, will be recognized at the receiver if they have enough discrimination features between each other. The control unit of this module, which is located at data link layer, is used to optimize the spectral usage under low signal-to-noise ratio (SNR) by selecting the appropriate glossary at the physical that would reduce the bit error rate (BER). Some distinct features of the proposed scheme are as follows.

- In our previous work [10, 42], the modulation and encoding were performed by means of sinusoidal pattern envelope construction (SPEC) algorithm, which generates random symbols with unequal spacing. This has a key impact on the symbol recognition rate of the neural network. In other words, the neural network

fails to recognize the distorted symbols if at least two of them have common features. In this work, we proposed a new algorithm, called advanced sinusoidal pattern envelope construction (ASPEC) algorithm, which improve the recognition rate as well as BER. The main aspect of the proposed algorithm is carefully selecting the symbols that maximizes the minimum distance of the glossary dataset.

- As an advanced step that takes the benefit from the rapid development of embedded and wireless technology, the pattern-based cognitive communication system is realized with the Xilinx ZYNQ ZC706 evaluation board, a high performance Field Programmable Gate Arrays (FPGA) embedded platform. FPGAs, a key enabling technology for SDR implementation, have been extensively studied as an important hardware platform for SDR. FPGA is different from general-purpose and digital signal processing architectures because it allows for customizing the functions of the designed hardware. Throughout this article, we consider both theoretical and practical aspects and present the hardware implementation of PBCCS, demonstrating real-time system.

The content of this paper is organized as follows. Section 5.2 describes the transceiver architecture of the PBCCS. We then present the ASPEC algorithm, which is used to construct the optimal symbols in Section 5.3. The hardware system platform and the software platform are presented in Section 5.4. Experiments results and analysis are given in section 5.5. It also presents the performance of the transceiver using our hardware-in-the-loop testbed. The conclusion is made in Section 5.6.

5.2 Overview of PBCCS

PBCCS consists of two main parts, the transmitter and receiver. Basically, the system employs pattern-based encoding at the transmitter and a wavelet-preprocessed artificial neural network based decoder at the receiver. In this section, we describe briefly the individual parts of PBCCS.

5.2.1 How does PBCCS work?

Figure 5.1 shows the cross-layer architecture of the PBCCS. In this model, PBCCS software framework layer provides necessary system services and programming

support for implementing PBCCS. The physical layer includes the radio front-end, a hardware module that transmits and/or receives radio signals through an antenna.

On the transmitting side, a sequence of bits is received from the data link layer and the encoder should map each block of data to a specific symbol, $m(t)$. Essentially, the glossary controller, the cognitive controller of PBCCS, picks up one glossary from the glossary dataset according to CSI acquired at the receiver. Then, it matches each block of data with the corresponding baseband waveform from the selected glossary. At the physical layer, a digital-up converter (DUC) transfers the baseband signal to IF and the DAC that follows transform the samples to the analog domain. Next, the RF converter shifts the signal towards higher frequencies. Finally, the signal is amplified and directed to the antenna.

The RF front-end at the receiver acquires an analog waveform from the antenna, down converts it to a lower frequency, and then digitizes it into discrete samples before handing it to the PBCCS software framework. Upon receiving a frame from the physical layer, the main signal features are extracted by the discrete wavelet transform (DWT). Subsequently, the ANN is used to classify the signal features to obtain the transmitted block of data. The decoded bit streams are mapped to packets, which are pushed upwards to the data link layer.

5.2.2 How does the PBCCS encodes the data?

The encoding process involves two steps—i.e., the offline and online process. In the offline process, the glossary is constructed using SPEC algorithm 1, which will be described in the next section. Generally, the glossary is a set of 2^k symbols, where k is the number of represented bits, which presents one binary word with different length. The symbol characteristics that have been quantified are considered to be of relevance to glossary usability, such as the immunity to high-level SNR, efficient spectral usage and channel bandwidth. Therefore, the generated signal of each symbol has different waveforms that change over time in terms of amplitude, frequency and phase to meet the symbol characteristics. The aggregated glossaries form a dataset with varied characteristics.

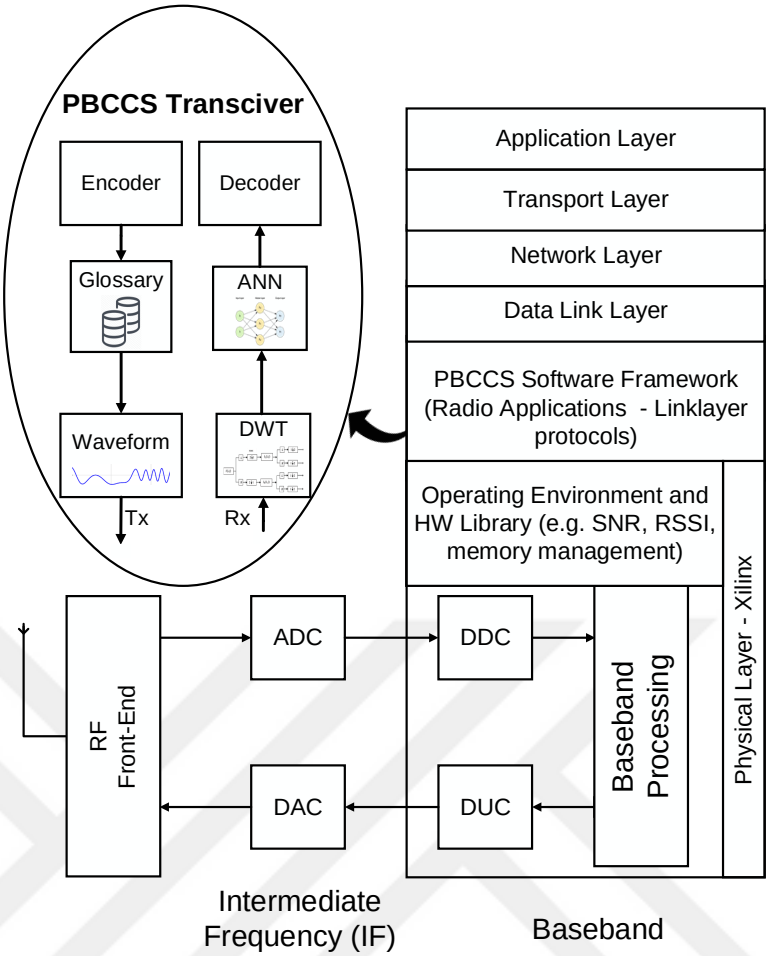


Figure 5.1 : The cross-layer structure of PBCCS with general software-defined layer.

In the online step, which is the glossary controller, the core component in the transmitter, selects the most appropriate glossary from the glossary dataset as part of the adaptation process. It takes the glossary information and channel spectral situation— i.e., the SNR value from the environment, as an input to determine the most proper glossaries set in the glossary space. Based on these values, it computes the maximum likelihood value by using a hysteresis function.

To summarize, with the guide of the glossary selector, the encoder maps each block of data to a specific symbol, $m(t)$. The transmitted symbol, $m(t)$, contains a sequence of waveform. According to the selected glossary, the symbol that matches the data index is selected and applied to the RF front-end.

5.2.3 How does a PBCCS receiver decode the frame?

The main modules of the receiver are the discrete wavelet transform unit and ANN, as illustrated in Figure 5.1. The aim of using ANN at the receiver is to predict the

original bits of the distorted received signal. In contrast to matched filters, the receiver does not apply any signal to estimate its parameter nor to construct a similar analog signal. Instead, it classifies the input samples to a known pattern, so that the correct bits can be inferred. The only requirement to successfully classify the symbols is learning the ANN weights from the constructed symbol (during the offline step).

5.2.3.1 Features extraction and reduction

One of the aspects of signal classification is the selection of proper classification features. The goal of feature extraction is to obtain a set of features that can discriminate different received signals. In this work, the discrete wavelet transform (DWT) [32, 94] is used to extract the signal features.

The discrete wavelet transform is a linear signal processing technique that transforms a signal $r(t)$ from the time domain to the "wavelet" domain—i.e., wavelet coefficients. A transformation from the time domain to the "wavelet" domain is analogous to the Fourier transform. The key difference between wavelet transform and Fourier transform is that wavelets are local in both time (via translation) and frequency (via dilation), whereas Fourier analysis is local only in frequency but not in time. Because the generated waveforms contain numerous non-stationary or transitory characteristics, which are often the most important parts of signals, Fourier analysis is unsuitable to describe such characteristics. Moreover, the received pattern signal can be represented by a compact form and hold most features that distinguish it from other patterns. As a result, the wavelet analysis is appropriate to capture the changes in the pattern's frequency over time and achieves better lossy compression, which dramatically reduces the size of ANN.

5.2.3.2 Recognition layer

After extracting the proper features from the received signal, classifying these patterns into appropriate classes is the final step to recognizing the symbol. In this work, the artificial neural network (ANN) [34, 35] is considered as a recognition layer to recover the transmitted data, and it forms the cognitive part of the PBCCS receiver.

The key reason for choosing the preprocessed wavelet with ANN is its high sensitivity for recognizing the amplitude, frequency and phase changes in the communication

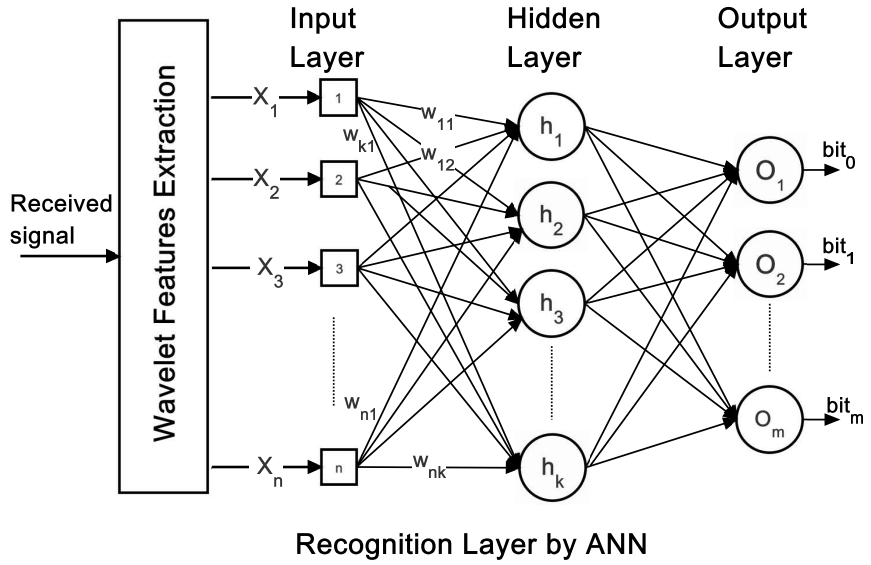


Figure 5.2 : Feed-forward ANN (FFNN) with one hidden layer and an output layer. The extracted features are fed into ANN with n input nodes, and k and m are the number of hidden and output neurons, respectively. The output bits are the bits that were recognized by the FFNN.

signal. The output that is produced by the ANN is decoded in the final stage into the correct bit sequence, as shown in Figure 5.2.

In this work, the most common ANN model, namely multilayer perceptron (MLP), is used. MLP is a type of feed-forward neural network (FFNN) model that maps the input data onto a set of appropriate outputs. It consists of at least three layers—i.e., the input layer, one or more hidden layers and an output layer. The network is fully connected from one layer to the next as a directed acyclic graph. Each neuron is capable of multiplying the inputs by its weight and summing up the results.

5.3 The Advanced Sinusoidal Pattern Envelope Construction (ASPEC) Algorithm

PBCCS has its own waveform construction technique and can only uses them. It uses an envelope pattern to modulate the information, which is a form of pulse shape. The generated symbol, which consists of several pattern, is important for recovering the information. Therefore, the central frequency of signal pattern can be adaptively shifted to any free communication channel.

5.3.1 Glossary construction

In PBCCS, the modulation is performed by using the advanced sinusoidal pattern envelope construction (ASPEC) algorithm, an optimized version of sinusoidal pattern envelope construction (SPEC) [10, 42]. One of the main drawbacks of the SPEC algorithm is the generation of random symbols with unequal spacing. This means that several symbols at the same glossary could have common features, which decreases the recognition rate of those symbols with features in common. The accuracy, which indicates the success rate of the recognition symbols, is significantly decreased.

ASPEC algorithm ensures that the spacing between symbols is maximum. In addition, it prevents unwanted extra spectral usage by employing quarter sinus form as the basis units for shift keying instead of linear shift. It guarantees that the signal's pattern ends at its initial point to alleviate the synchronization effect. The key difference between these two algorithms is how they construct the routes from the initial point to the end.

ASPEC has two essential parameters—namely, "depth" and "level", which are similar to SPEC's parameters. Depth determines the length of the pattern in terms of time—i.e., number of periods. Meanwhile, level identifies a value for each feature of the signal pattern. It represents the maximum and minimum values of any signal characteristics at depth i, e.g. amplitude (A), frequency (F) or phase (P). All possible outcomes of ASPEC algorithm are due to the changes in A, F and P features of the signal along the depth's values and levels. For instance, Figure 5.3.(a) shows the envelope combinations of ± 3 levels (the vertical axis) and 6 depth layers (the horizontal axis). The minimum and maximum levels are mapped to the minimum and maximum signal's feature. If 1 MHz and 2 MHz are the minimum and maximum frequency of the signals, then minimum level is mapped to 1 MHz and the maximum is mapped 2 MHz. The in-between levels are equally distributed between them. Similarly, Figure 5.3.(b) and 5.3.(c) show the same envelop with ± 5 and ± 7 levels, respectively.

Algorithm 1 Signal Generation using ASPEC.

Input:
A_{min}, A_{max} : The minimum and maximum amplitude
F_{min}, F_{max} : The minimum and maximum frequency
P_{min}, P_{max} : The minimum and maximum phase
$l \leftarrow$ Number of levels
$d \leftarrow$ Number of depths
$k \leftarrow$ Number of bits
$s \leftarrow$ Total number of samples per symbol
$pr \leftarrow$ Number of periods
Output: $k-$bit glossary

1: (Step 1) CreateRoutes(l, d)
2: $K_{max} \leftarrow$ Number of routes
3: CheckOrthogonality()
4: Sort the result
5: routes $\leftarrow$ Select the most uncorrelated 2^k routes
6: (Step 2) features $\leftarrow$ CreateSignalFeatures(routes, A_{min}, A_{max}, F_{min}, $F_{max}, P_{min}, P_{max}, k, s$)
7: $F_{avg} = (F_{min} + F_{max}/2)$
8: (Step 3) ConstructPattern(features,F_{avg}, s, pr)
9: **return** $k-$bit Glossary

5.3.2 ASPEC algorithm for glossary construction

ASPEC Algorithm performs the glossary construction in three steps. The first step generates all possible routes that alternates between the depth and level, then selects the top 2^k uncorrelated routes. The second step creates the signal features with the corresponding routes. The last step, combines the features to construct the signals. Algorithm 1 describes the ASPEC steps in more detail.

CreateRoutes(l, d) creates all possible routes from $(0,0)$ to $(0, d+1)$ by alternating between all possible levels and depths value. It should start from level 0 and end at the same level, 0. Any route is represented by a vector of length equal to d and three possible values, e.g. $-1, 0$ or 1, where -1 indicates the level is decreased, 1 indicates the level is increased and 0 if the level remains the same as the previous level. For example, the red route in Figure 5.3.(a) is indicated by $[-1,1,-1,1,-1,1]$ and the blue one is indicated by $[1,-1,0,0,1,-1]$. Because the glossary should contain the most uncorrelated signal envelopes, *CheckOrthogonality()* function is used to calculate the orthogonality between each route.

After that, 2^k routes are selected to create the $k-$bit glossary with 2^k patterns. *CreateSignalFeatures()* takes the minimum and maximum value of the signal's amplitude, frequency, phase and the number of samples. It returns the feature matrix of each property with respect to each sample and bit. Algorithm 2 shows one way to construct the signal features. Finally, *ConstructPattern(features,*F_{avg}, s, pr*)* constructs each pattern as follows:

$$m_i(j) = a_j * cos(2\pi f_j(j * pr/(s * F_{avg})) + \phi_j), 1 \leq j \leq s \tag{5.1}$$

where $m_i(j)$ is the j^{th} sample of the i^{th} pattern; a_j ,f_j and ϕ_j are the amplitude, frequency and phase values at the j^{th} sample, respectively; pr is the number of period;s is the total number of samples per symbol; and F_{avg} is average frequency.

According to equation 5.1, each block of k input data bits $d_i, i = 1, 2, ..., 2^k$ is mapped to a pattern $m_i(t)$. All symbols, $m_i(t)$, are combined to form a k-bit glossary space. Figure 5.4.(a) and Figure 5.4.(b) illustrate a 3-bit glossary space, containing eight signal patterns with $pr = 3$ and 5 periods, respectively.

It is worth noting that there are some prerequisites to construct signal patterns (waveform) of a glossary. In terms of a fundamental necessity of the RF communication, the constructed signal patterns should satisfy the following three conditions:

1. Zero DC component: average of each signal pattern and the series of patterns must be zero, so that the RF energy can efficiently be transferred.

2. Continuity: the first and second derivatives of the signal patterns and the consequent patterns must be continues in order to prevent non-transferrable harmonics beyond the pass-band.

3. Bandwidth limitation: spectral distribution of the generated signal should ideally be inside the predefined frequency interval.

5.4 Implementation

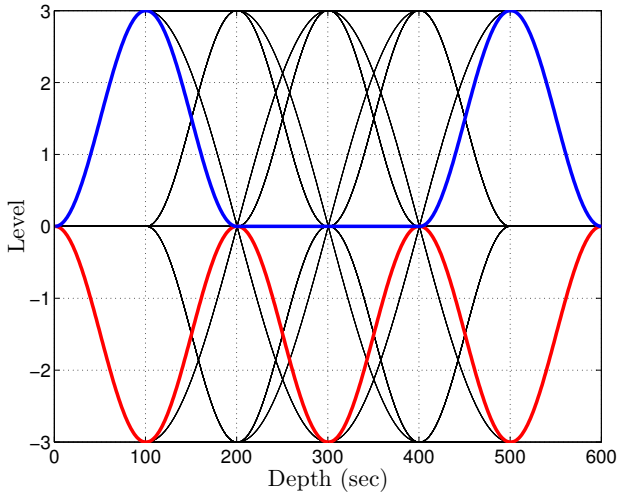
In this section, we briefly describe the current implementation of PBCCS that is based on FPGA. Basically, we describe our hardware prototype and software stack.

Algorithm 2 Create Signal Features.

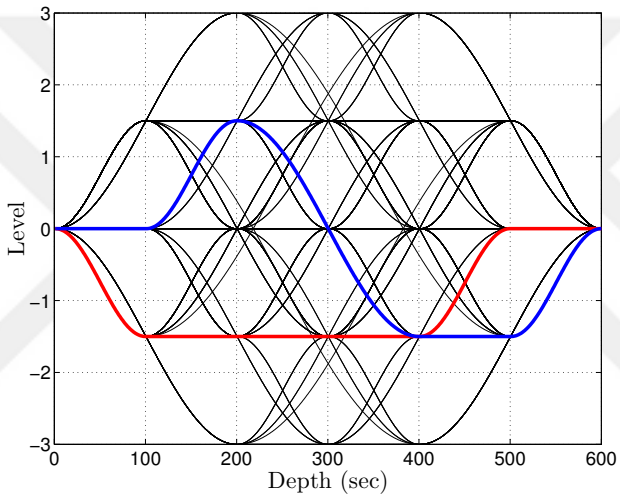
Input:
f_{min}, f_{max} : The minimum and maximum frequency
$s \leftarrow$ Samples/Symbol
$l \leftarrow$ Number of levels
$d \leftarrow$ Number of depths
$k \leftarrow$ Number of bits
$routes \leftarrow$ orthogonal routes
Output: The waveforms of the required feature.

for each $r \in routes$ **do**
    $w \leftarrow 0$
    for $J = 1$ to d **do**
        if routes$(i, j) = -1$ **then**
            if routes$(i, j+1) \neq -1$ **then**
                SubSignal = semi_decrease()
            else
                SubSignal = full_decrease()
            end if
        else if routes$(i, j) = 0$ **then**
            if routes$(i, j+1) \neq 0$ **then**
                SubSignal = semi_constant()
            else
                SubSignal = full_constant()
            end if
        else if routes$(i, j) = 1$ **then**
            if routes$(i, j+1) \neq 1$ **then**
                SubSignal = semi_increase()
            else
                SubSignal = full_increase()
            end if
        end if
        append *SubSignal* to w
    end for
    $waveform(r) \leftarrow w$
end for

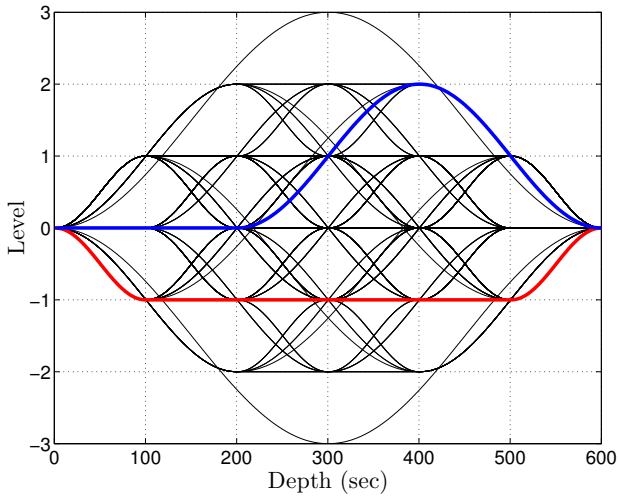
We utilized a Xilinx ZYNQ ZC706 evaluation board to realize and verify the algorithm/concept of PBCCS, described in the previous sections. Furthermore, the re-programmability of this board allows for realizing the design without incurring significant re-construction costs.



(a) 3 levels and 99 routes.

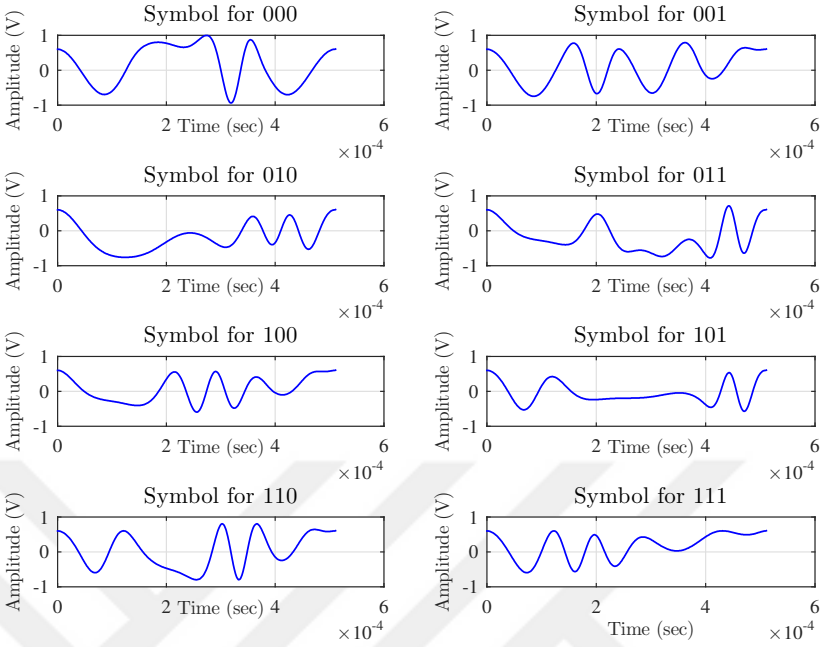


(b) 5 levels and 139 routes.

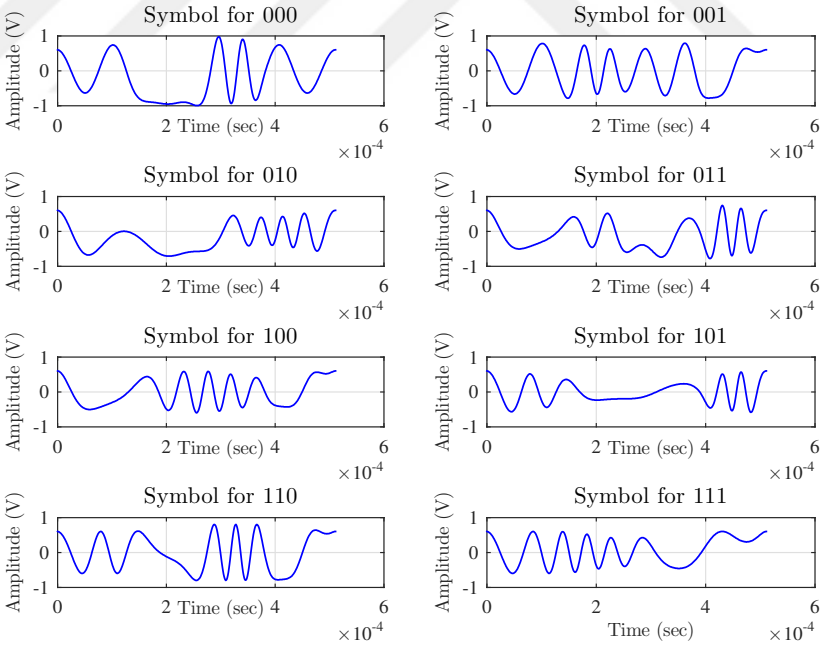


(c) 7 levels and 141 routes.

Figure 5.3 : ASPEC with all possible routes for depth equal to 6 and different levels.



(a) with number of period, $pr = 3$.



(b) with number of period, $pr = 5$.

Figure 5.4 : The 3-bit glossary space with 5 levels and depth is set to 6. The signal's amplitude is in [0.2 , 1] V, frequency is in $[0.1, 1]$ MHz, and phase shift is [-5 , 5] rad.

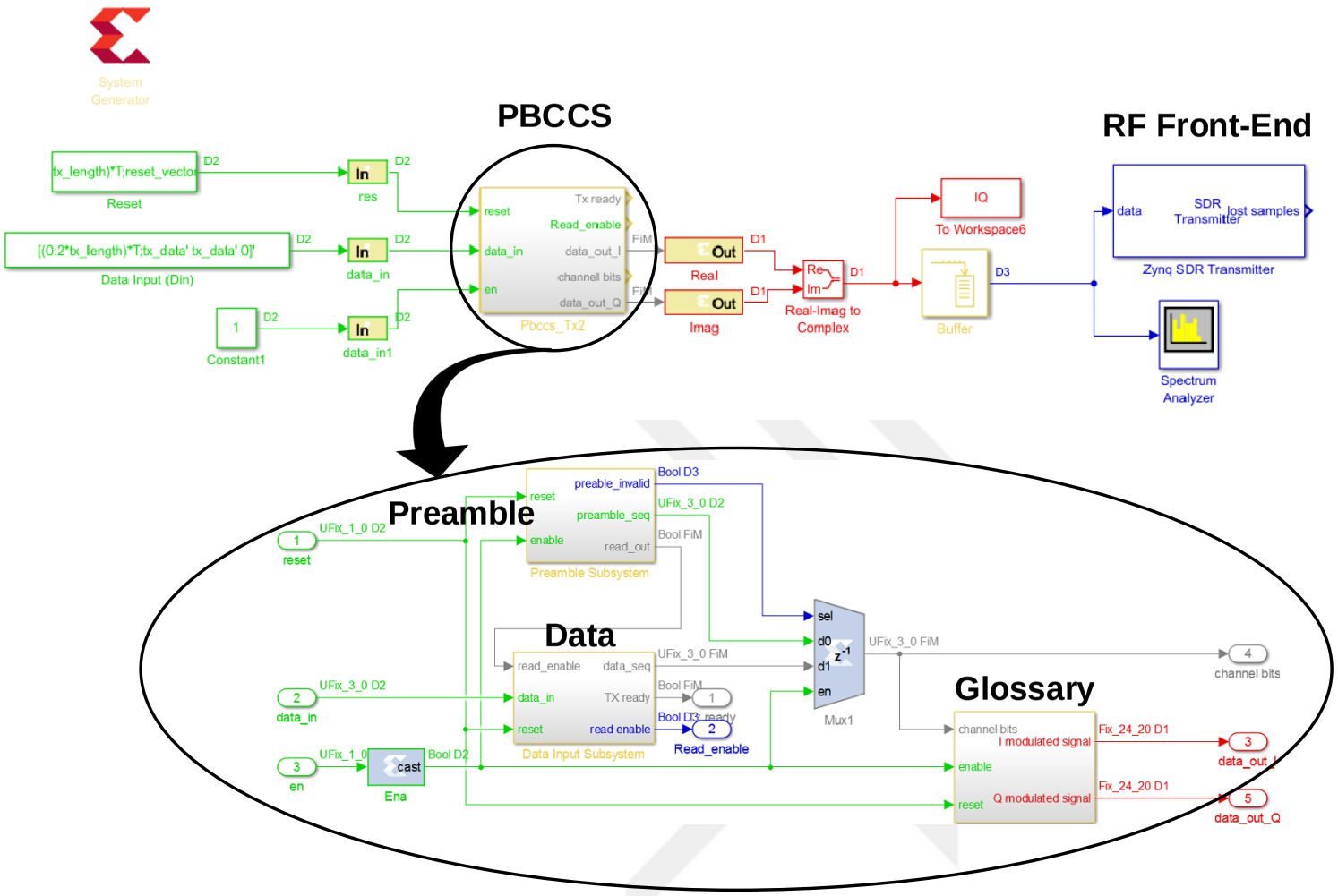


Figure 5.5 : The Simulink block diagram of the PBCCS transmitter.

5.4.1 Hardware platform

The transceiver model was implemented on top of ZYNQ-7000 SoC [88]. The hardware platform is mainly comprised of a FPGA chip Xilinx ZYNQ-7000 and an integrated transceiver chip AD9361 [95]. The ZYNQ-7000 is a programmable System-on-Chip (SoC) that supports processing system (PS) with a dual-core ARM Cortex A9 MPCore and a Xilinx programmable logic (PL) in a single device. The general software-defined layers of PBCCS and its main blocks is illustrated in Figure 5.1.

To simplify the design of radio frequency, we used AD-FMCOMMS3 from Analog Devices, which involves an AD9361 as a high performance and a high integrated RF Agile Transceiver. Due to its programmability and wide band capability, it is suitable for a broad range of transceiver applications. It is composed of up-converter, down-converter, filter, I/Q modulator, I/Q demodulator, ADC and DAC. The AD9361 can operate between 70 MHz to 6.0 GHz, which covers most licensed and unlicensed spectrum. The sampling channel bandwidth varies from 200 kHz to 56 MHz.

5.4.2 Software platform

The PBCCS transceiver was modeled by using Xilinx System Generator for DSP (SysGen) [96], which is part of the Vivado System Edition Design Suite [97]. SysGen is a high-level software tool that enables the use of MATLAB/Simulink environment to create and verify hardware designs for Xilinx FPGAs. We prefer to use SysGen because it is a better tool for proof of concept and functional verification.

Figure 5.5 shows the SysGen Simulink model of PBCCS transmitter. The preamble is 12-bit length and is used for message synchronization, which is 128-bit length long. Data input block is a FIFO model that reads the data and buffered them until enable signal (en) from the preamble is activated.

The glossary model is composed of a memory that contains the 3-bit glossary signals— i.e., 8 signals, each of 24-bit samples. To reduce the number of required memories, we decided to use one memory that merges between the I and Q signals of $m(t)$— i.e., 8

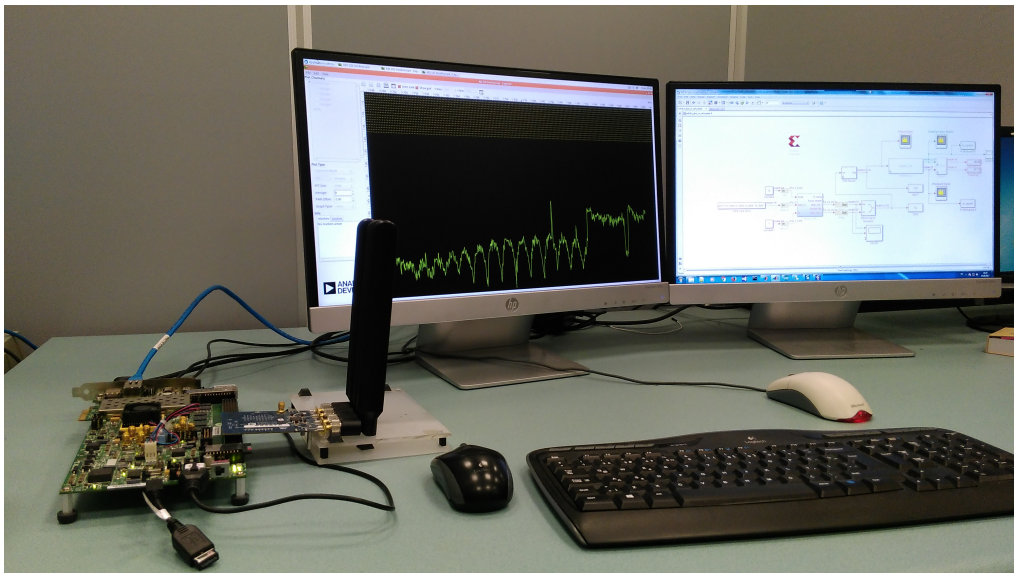


Figure 5.6 : Experimental testbed setup of Xilinx ZC706 and AD9361 board to implement the PBCCS transmitter.

was needed. This memory is indexed by 3-bit of the data stream. Then, I and Q signals are separated before signal transmission.

The entire transmitter model is shown in Figure 5.5. The transmitter starts by mapping digital data onto the corresponding symbols. Then, it generates the corresponding baseband IQ signals. These baseband waveforms are modulated for RF transmission at the AD9361 blocks, where a digital up-converter (DUC) is used for up-sampling conversion— i.e., from 7.68 MHz to 780 MHz.

5.5 System Evaluations

Figure 5.6 shows the test setup of the system, which mainly includes Xilinx ZC706 and AD9361 board. Figure 5.7 illustrates the experiment setup used for transmitting and receiving short messages, consisting of one SDR node. We emphasize here that the 3-bit glossary at the transmitter is stored in a high-speed memory and indexed by 3-bit input.

5.5.1 Message format

This transmitter is used to send short messages. Each message contains 28 characters, and each character is represented by 7 bits. The total frame-length is 213 bits, where the first 12 bits are used for signaling, as shown in Figure 5.8.

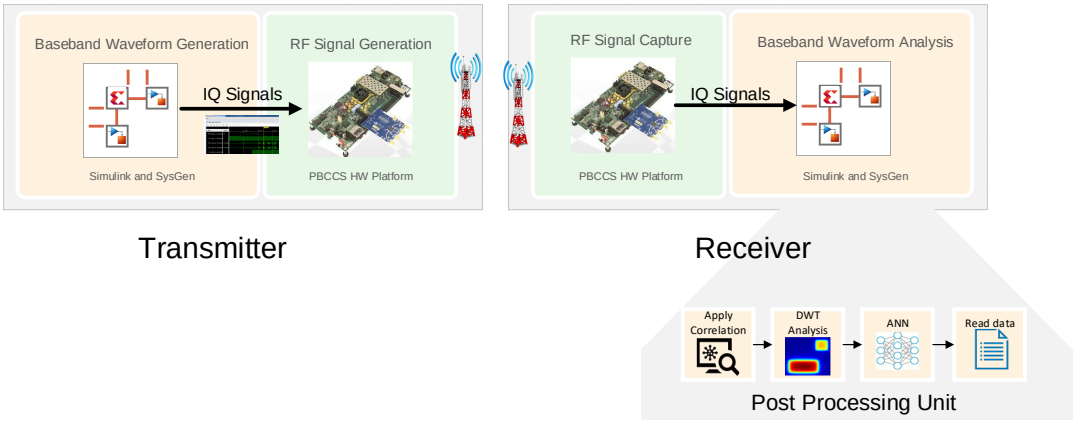


Figure 5.7 : The transceiver Model.

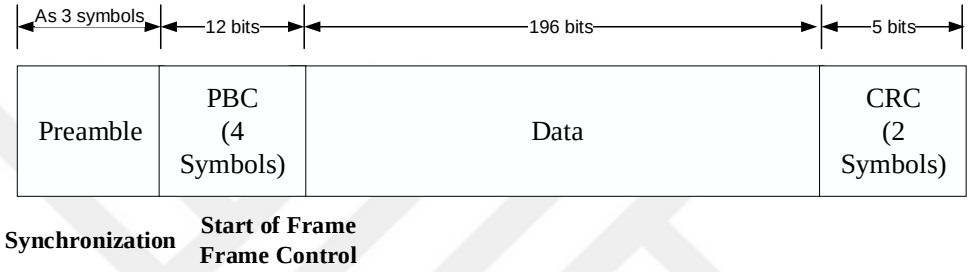


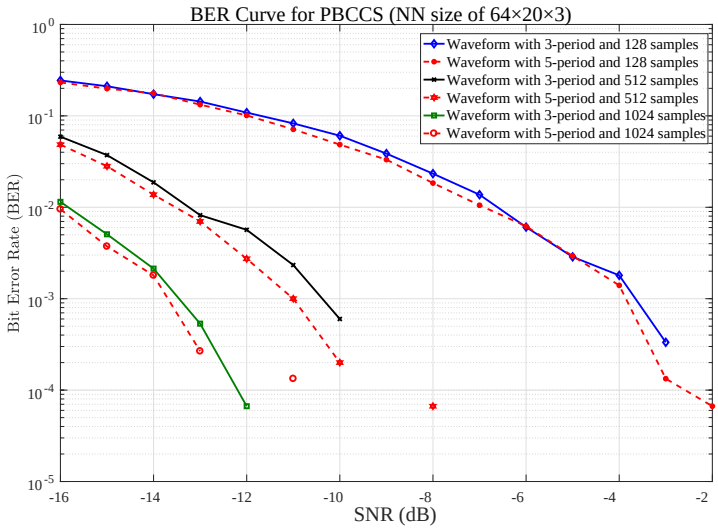
Figure 5.8 : The frame format of the transmitted message.

5.5.2 System performance

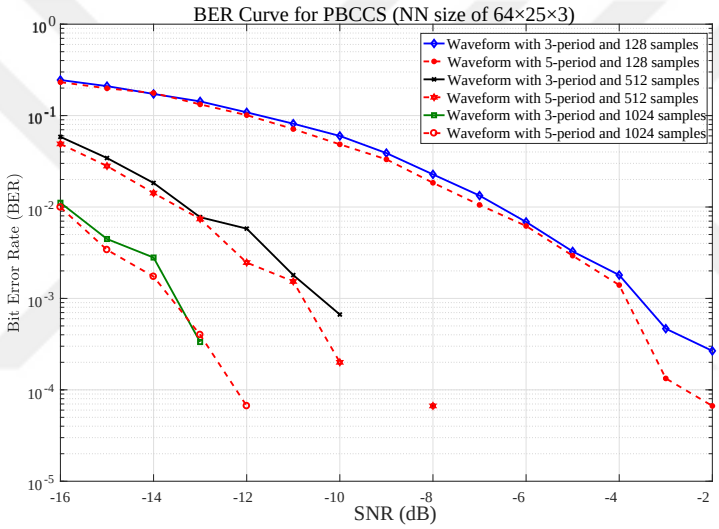
Table 5.2 presents the resource utilization of the PBCCS transmitter. As it can be noticed, 6 Block RAMs are used. One of them is used to store the symbols and the rest are used for controlling the system. The maximum operating frequency of the FPGA is 342.2 MHz.

5.5.3 Classification performance

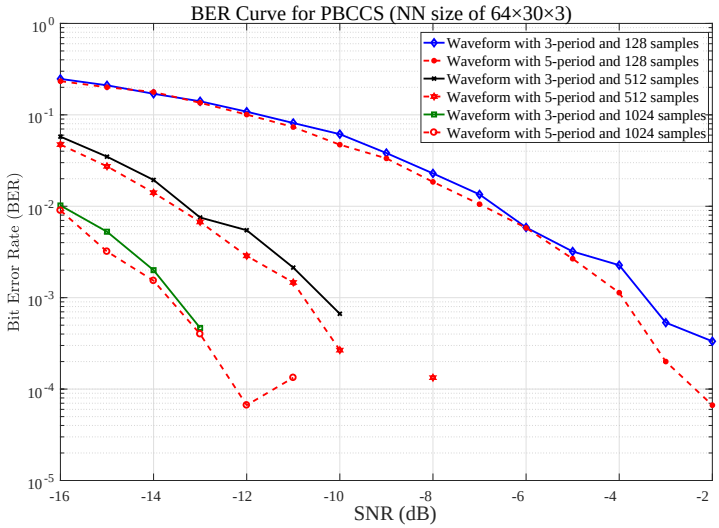
The PBCCS transmitter contains a 3-bit glossary to encode digital data into specific analog symbol. In this part, a comparison between the BER of different glossaries is presented. We studied the impact of increasing the signal periods and the number of samples on the spectrum. The BER performance of the PBCCS system is evaluated using the parameters illustrated in Table 5.1. We have selected 780 MHz as the frequency carrier (f_c). Figure 5.9 shows that as the number of samples and periods (signal features) increases, BER at the receiver is improved. This means that symbol recognition rate is also significantly improved.



(a) BER with 20 neurons.



(b) BER with 25 neurons.



(c) BER with 30 neurons.

Figure 5.9 : The effect of increasing the sampling rate and the number of periods on BER with different neurons at the hidden layer.

Table 5.1 : The main system settings that are used by PBCCS transceiver.

Parameter	Value
Signal BW	$\leq$ 2 MHz
Sampling Frequency (F_s)	7.6 MHz
IF sampling frequency	122.8 MHz
Carrier Frequency	780 MHz
Buffer size	16384 samples
No. of samples	16384 samples
No. of samples/symbol	$64, 128, 256, 512 and 1024$ samples
Frame Length	213 bits
Message Length	196 bits

Table 5.2 : Resource utilization and system performance of 3-bit glossary on Xilinx Zynq-7000 board.

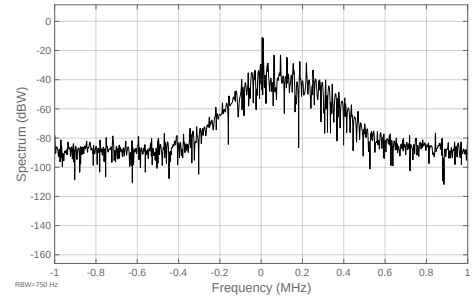
Resource	Utilization	Available	%
Number of Slice Registers	213	437600	0.05
Number of Slice LUTs	105	218600	0.05
Number of occupied Slices	65	54650	0.0012
Number of BRAM	6	545	1.1
Number of DSP	0	900	0
Worst neg. slack (ns)		13.078	
Max. operating freq (MHz)		342.2	
Data path delay (ns)		2.85	
Estimated Power (mW)		3	

5.5.4 Spectrum analysis

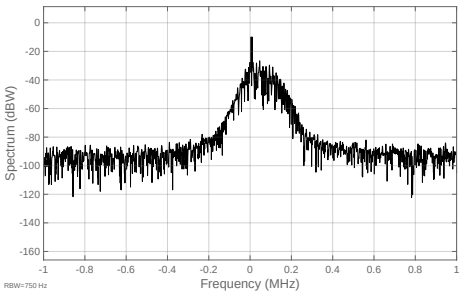
Figure 5.10 shows the spectrum of 64-, 128-, 256- and 512-sample symbol, respectively. As the number of samples is increasing, the signal power is observed to be concentrating at the center. This means, the noise (harmonics) effect on the near bands is decreasing. However, increasing the number of samples reduces the data rate from 5.7562 Mbit/s to 0.7195 Mbit/s. Table 5.3 compares the observed occupied bandwidth between 3- and 5-period 3-bit glossary.

5.5.5 System performance comparison

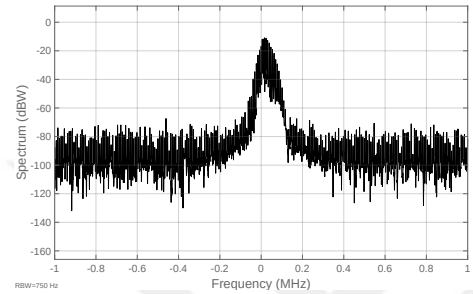
PBCCS was designed to increase the link spectral efficiency (LSE) by constructing a secure and non-periodic communication signals. Table 5.4 presents a comparison between the PBCCS and standard communication systems. It is clear that LSE of PBCCS is greater than the LSE of 2G and WCDMA. In contrast, this value is smaller



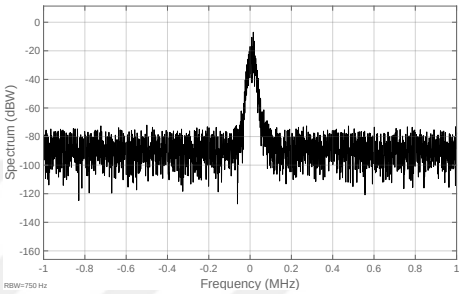
(a) spectrum with 64 samples/symbol.



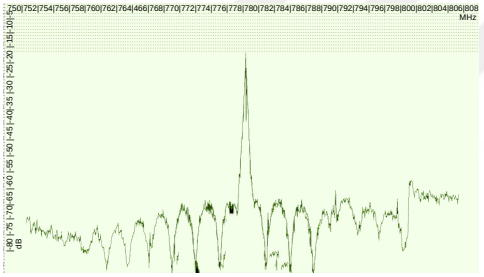
(b) spectrum with 128 samples/symbol.



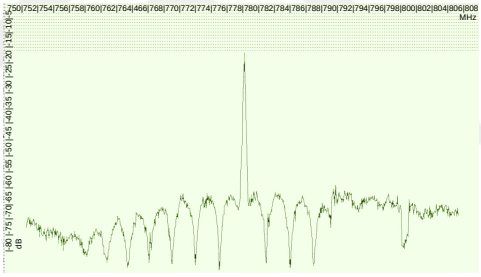
(c) spectrum with 256 samples/symbol.



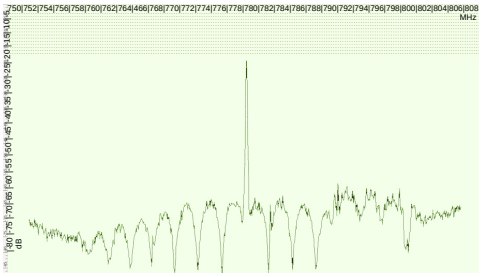
(d) spectrum with 512 samples/symbol.



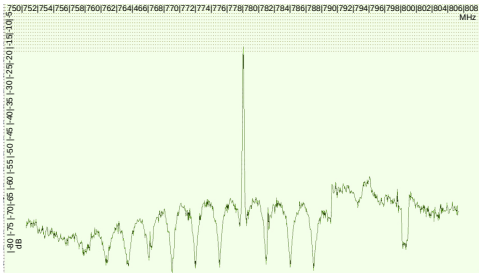
(e) spectrum with 64 samples/symbol.



(f) spectrum with 128 samples/symbol.



(g) spectrum with 256 samples/symbol.



(h) spectrum with 512 samples/symbol.

Figure 5.10 : The 3-bit glossary baseband spectrum as displayed by Simulink spectrum analyzer, using 3-period signals per symbol, each of (a) 64, (b) 128, (c) 256, (d) 512 samples/Symbol. The transmitted spectrum as displayed by spectrum analyzer, centered at 780 MHz and using 3-period signals per symbol, each of (e) 64, (f) 128, (g) 256, (h) 512 samples/Symbol.

Table 5.3 : Observed bandwidth of 3-bit glossary on Xilinx Zynq-7000 board.

	Number of samples			
	64	128	256	512
Data Rate (Mbit/s)	5.7562	2.8781	1.4391	0.7195
3 periods BW (MHz)	2	1.3	0.6	0.4
LSE (bit/s/Hz)	2.878	2.3984	2.3985	1.7988
5 periods BW (MHz)	2.2	1.2	0.6	0.4
LSE (bit/s/Hz)	2.6165	2.2139	2.3985	1.7988

Table 5.4 : LSE Comparison of Standard Commercial Communication System.

	PBCCS	GSM	WCDMA	HSPA$^+$	LTE
BW (MHz)	0.4	0.2	5	5	20
Datarate (Mbps)	0.7195	0.104	0.384	42	81.6
LSE (bit/s/Hz)	1.7988	0.52	0.0768	4.08	3.75

than the LSE of HSPA$^+$ and LTE because these systems applies different technicalities to increase the LSE values such as OFDMA and MIMO.

5.6 Conclusion

In this work, we have introduced an improved glossary design that yields reliable classification performance for commercial communication applications as well as a considerable increase of the system's spectral efficiency. We have described the implementation of the PBCCS transceiver on FPGA. In addition, we have considered improving the receiver recognition rate under different SNR levels.

The performance of transmitter was evaluated on Xilinx ZYNQ ZC706 development kit. We addressed the effect of increasing the number of samples per symbol, which has a substantial influence on the overall performance of the spectrum. The main findings show that as the number of samples increase, both spectral efficiency and the recognition rate improved. Future work will focus on integrating the transmitter and receiver on the same board, verifying and measuring the real-time performance of the whole system.

6. CONCLUSIONS AND FURTHER WORK

This dissertation described the pattern-based cognitive communication system that applies a wavelet preprocessed neural network for the detection of the received signal. In this work, we have proposed the first prototype of pattern-based cognitive communication system as well as experimental evaluations. The robustness of the proposed system against low-SNR in the communication channel have also been evaluated. Additionally, we have addressed the development of discrete wavelet transform model that is used to preprocess the received signal. The essentials of the RNS and DAA multiplierless implementation and their properties, advantages, and disadvantages were surveyed and studied. With a brief discussion on the various implementation of DWT, we concluded RNS to be the best implementation for the proposed receiver.

Through this dissertation, the core development of PBCCS was done through the use of Simulink, MATLAB scripts and Xilinx System Generator for DSP (SysGen) modules glued within the Simulink environment before being ported to Xilinx EDK/SDK environment. Details on how the environment has been setup was also presented. We have developed and validated the proposed scheme, and have demonstrated an implementation of the pipeline on the Xilinx ZYNQ ZC706 FPGA.

The results have highlighted some aspects of PBCCS implementations. Following is the major key featured summary of this research work:

The proposed model increases the spectral efficiency by constructing a secure and non-periodic communication signals. In addition, the proposed architecture minimizes the bit error rate (BER) through optimized signal patterns that are decoded solely by DWT preprocessed signals and artificial neural network. Finally, PBCCS has better error performance over traditional modulation schemes.

6.1 Practical Application of This Study

Having validated PBCCS's design flexibility and control through the above experiments, this dissertation derives several key applications.

The most obvious applications that can benefit from PBCCS are spectrum-aware communication applications. According to the channel and receiver status, these applications adapt their behavior by changing the working glossary. The provided cross-layer design supports the communications between the physical and data link layers. The cognitive engine of PBCCS, which is located within the data link layer, is able to switch between the available glossaries to optimize the spectral usage under various SNR levels.

Another application describe a novel physical layer based technique for securing wireless communication between the transmitter and receiver. The system involves a glossary that works as database of different symbols. The receiver will be able to decode the received symbols if and only if it has the enough knowledge of the glossary's features. According to our experience, intruders would be unable to recover symbols without any prior knowledge of the used glossary.

Last but not the least, PBCCS provides a candidate solution for jamming attacks. In general, jammers send a strong signal that interfere any signal in the surrounding area and prevents the use of existing channel. Because PBCCS is based on a cross-layer architecture, the transceiver switches to new glossary with higher recognition ability at the receiver.

6.2 Future Research

The possibilities of future research surrounding pattern-based cognitive communication are both exciting and wide. Here, we hereby highlight the following as basic guidelines for any future work in this field.

- The problem of multi-user is still an open issue. The glossary selector can be extended by a controller to manage the glossary sets among the users. Also, the receiver should be notified with which respective glossary to be used and with which respective user is going to be using it.

- In order to meet the time line and dissertation scope, we concentrated on designing PBCCS. Although implementing PBCCS has value on its own, we expected a much more detailed performance evaluation of the system. In particular, the current design and experiments are limited to spectral performance of the transmitter. Leaving open a series of questions including the multi-path and Doppler effect.

- Motivated by recent advances and the remarkable success of deep learning, especially convolutional neural networks (CNN), the performance of PBCCS‘s receiver can be greatly improved. The main issue is that the learning process will consume more time and require high performance machines.

- While spectrum sensing enables cognitive radio systems, the adaptive transmission over white spaces is critical for capacity increase and effectiveness of spectrum sharing. On the physical layer, it would be interesting to explore new signaling strategies for different regime of spectrum availability. In addition, a design of modulation schemes that can meet co-channel and adjacent channel interference without external filtering is desirable for a flexible digital implementation.

REFERENCES

[1] **Dobkin, D.M.** (2004). *RF Engineering for Wireless Networks: Hardware, Antennas, and Propagation (Communications Engineering)*, Newnes, MA, USA.

[2] **Proakis, J.G. and Salehi, M.** (2008). *Digital Communication*, McGraw-Hill Education, New York, NY, USA, 5th edition.

[3] **Mitola, J. and Maguire, G.Q.** (1999). Cognitive Radio: Making Software Radios More Personal, *Personal Communications, IEEE*, *6*(4), 13–18.

[4] **Haykin, S.** (2005). Cognitive Radio: Brain-Empowered Wireless Communications, *Selected Areas in Communications, IEEE Journal on*, *23*(2), 201–220.

[5] **Haykin, S.** (2012). *Cognitive Dynamic Systems: Perception-action Cycle, Radar and Radio*, Cambridge University Press, New York, NY, USA.

[6] **Yucek, T. and Arslan, H.** (2009). A Survey of Spectrum Sensing Algorithms for Cognitive Radio Applications, *Communications Surveys Tutorials, IEEE*, *11*(1), 116–130.

[7] **Akyildiz, I.F., Lee, W.Y., Vuran, M.C. and Mohanty, S.** (2006). NeXt Generation/Dynamic Spectrum Access/Cognitive Radio Wireless Networks: A Survey, *The International Journal of Computer and Telecommunications Networking*, *50*(13), 2127–2159, `http://dx.doi.org/10.1016/j.comnet.2006.05.001`.

[8] **Shannon, C.** (1948). A mathematical Theory of Communication, *Bell System Technical Journal, The*, *27*(3), 379–423.

[9] **Ustundag, B. and Orcay, O.** (2008). Pattern Based Encoding for Cognitive Communication, Cognitive Radio Oriented Wireless Networks and Communications, 2008. CrownCom 2008. 3rd International Conference on, pp.1–6.

[10] **Ustundag, B. and Orcay, O.** (2011). A Pattern Construction Scheme for Neural Network-Based Cognitive Communication, *Entropy*, *13*(1), 64–81, `http://www.mdpi.com/1099-4300/13/1/64`.

[11] **Mitola, J.** (2000). *Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio*, (Ph.D. thesis), Royal Institute Technology (KTH), Stockholm, Sweden, http://www.diva-portal.org/smash/get/diva2:8730/FULLTEXT01.pdf.

[12] **Ahad, N.**, **Qadir, J. and Ahsan, N.** (2016). Neural networks in wireless networks: Techniques, applications and guidelines, *Journal of Network and Computer Applications*, *68*, 1 – 27, http://www.sciencedirect.com/science/article/pii/S1084804516300492.

[13] **Abbas, N.**, **Nasser, Y. and Ahmad, K.E.** (2015). Recent advances on artificial intelligence and learning techniques in cognitive radio networks, *EURASIP Journal on Wireless Communications and Networking*, *2015*(1), 174, https://doi.org/10.1186/s13638-015-0381-7.

[14] **He, A.**, **Bae, K.K.**, **Newman, T.**, **Gaeddert, J.**, **Kim, K.**, **Menon, R.**, **Morales-Tirado, L.**, **Neel, J.**, **Zhao, Y.**, **Reed, J. and Tranter, W.** (2010). A Survey of Artificial Intelligence for Cognitive Radios, *Vehicular Technology, IEEE Transactions on*, *59*(4), 1578–1592.

[15] **Alshawaqfeh, M.**, **Wang, X.**, **Ekti, A.R.**, **Shakir, M.Z.**, **Qaraqe, K. and Serpedin, E.** (2015). A Survey of Machine Learning Algorithms and Their Applications in Cognitive Radio, M. Weichold, M. Hamdi, M.Z. Shakir, M. Abdallah, G.K. Karagiannidis and M. Ismail, editors, Cognitive Radio Oriented Wireless Networks: 10th International Conference, CROWNCOM 2015, Doha, Qatar, April 21-23, 2015, Revised Selected Papers, Springer International Publishing, Cham, pp.790–801.

[16] **Bkassiny, M.**, **Li, Y. and Jayaweera, S.K.** (2013). A Survey on Machine-Learning Techniques in Cognitive Radios, *IEEE Communications Surveys Tutorials*, *15*(3), 1136–1159.

[17] **Fehske, A.**, **Gaeddert, J. and Reed, J.** (2005). A new Approach to Signal Classification Using Spectral Correlation and Neural Networks, New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on, pp.144–150.

[18] **Baban, S.**, **Denkoviski, D.**, **Holland, O.**, **Gavrilovska, L. and Aghvami, H.** (2013). Radio Access Technology Classification for Cognitive Radio Networks, Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on, pp.2718–2722.

[19] **Baban, S.**, **Holland, O. and Aghvami, H.** (2013). Wireless Standard Classification in Cognitive Radio Networks Using Self-Organizing Maps, Wireless Communication Systems (ISWCS 2013), Proceedings of the Tenth International Symposium on, pp.1–5.

[20] **lin Zhu, X.**, **an Liu, Y.**, **Weng, W.W. and Yuan, D.M.** (2008). Channel Sensing Algorithm Based on Neural Networks for Cognitive Wireless Mesh Networks, Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on, pp.1–4.

[21] **Zhou, L. and Man, H.** (2013). Wavelet Cyclic Feature Based Automatic Modulation Recognition Using Nonuniform Compressive Samples, Vehicular Technology Conference (VTC Fall), 2013 IEEE 78th, pp.1–6.

[22] **Abdelreheem, M. and Helmi, M.** (2012). Digital Modulation Classification through time and frequency domain features using Neural Networks, Telecommunications (BIHTEL), 2012 IX International Symposium on, pp.1–5.

[23] **Popoola, J. and Van Olst, R.** (2011). Application of Neural Network for Sensing Primary Radio Signals in a Cognitive Radio Environment, AFRICON, 2011, pp.1–6.

[24] **Attar, A.**, **Sheikhi, A. and Zamani, A.** (2004). Communication System Recognition by Modulation Recognition, J. de Souza, P. Dini and P. Lorenz, editors, Telecommunications and Networking - ICT 2004, volume 3124 of *Lecture Notes in Computer Science*, Springer Verlag, Berlin Heidelberg, Germany, pp.106–113.

[25] **Walenczykowska, M. and Kawalec, A.** (2016). Type of Modulation Identification Using Wavelet Transform and Neural Network, *The Journal of Polish Academy of Sciences*, *64*(1), 257 pp.–261.

[26] **Dahap, B.I.**, **HongShu, L. and Ramadan, M.** (2014). Simple and Efficient Algorithm for Automatic Modulation Recognition for Analogue and Digital Signals, B. Zhang, J. Mu, W. Wang, Q. Liang and Y. Pi, editors, The Proceedings of the Second International Conference on Communications, Signal Processing, and Systems, volume 246 of *Lecture Notes in Electrical Engineering*, Springer International Publishing, Switzerland, pp.345–357.

[27] **Alzaq, H. and Ustundag, B.** (2015). Wavelet Preprocessed Neural Network Based Receiver for Low SNR Communication System, European Wireless 2015; 21th European Wireless Conference, Proceedings of, pp.1–6.

[28] **Ebrahimzadeh, A. and Ghazalian, R.** (2011). Blind Digital Modulation Classification in Software Radio Using the Optimized Classifier and Feature Subset Selection, *Eng. Appl. Artif. Intell.*, *24*(1), 50–59, `http://dx.doi.org/10.1016/j.engappai.2010.08.008`.

[29] **Avci, E.**, **Hanbay, D. and Varol, A.** (2007). An expert Discrete Wavelet Adaptive Network Based Fuzzy Inference System for digital modulation recognition, *Expert Systems with Applications*, *33*(3), 582 – 589, `http://www.sciencedirect.com/science/article/pii/S0957417406001837`.

[30] **Hassanpour, S.**, **Pezeshk, A.M. and Behnia, F.** (2015). A robust algorithm based on wavelet transform for recognition of binary digital modulations, 2015 38th International Conference on Telecommunications and Signal Processing (TSP), pp.508–512.

[31] **Digdarsini, D.**, **Kumar, M.**, **Khot, G.**, **Ram, T.V.S. and Tank, V.K.** (2014). FPGA implementation of automatic modulation recognition system for advanced SATCOM system, 2014 International Conference on Signal Processing and Integrated Networks (SPIN), pp.464–469.

[32] **Mallat, S.** (2008). *A Wavelet Tour of Signal Processing: The Sparse Way*, Academic Press, Philadelphia, PA, USA, 3rd edition.

[33] **Daubechies, I.** (1992). *Ten Lectures on Wavelets*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

[34] **Haykin, S.** (2008). *Neural Networks and Learning Machines*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 3rd edition.

[35] **Hassan, Y.F.** (2011). Rough Sets for Adapting Wavelet Neural Networks as a New Classifier System, *Applied Intelligence*, *35*(2), 260–268, http://dx.doi.org/10.1007/s10489-010-0218-3.

[36] **4DSP**, Design and System integration for Digital Signal Processing, 4DSP - FMC150, available at http://www.4dsp.com/FMC150.php/.

[37] **Xilinx Inc.**, Virtex-6 FPGA ML605 Evaluation Kit, available at http://www.xilinx.com/products/boards-and-kits/ek-v6-ml605-g.html.

[38] **Prothero, J.** (2012). The Shannon Law for Non-Periodic Channels, **Technical ReportR-2012-1**, Astrapi Corporation, Washington, D.C.

[39] **Peled, A. and Liu, B.** (1974). A New Hardware Realization of Digital Filters, *Acoustics, Speech and Signal Processing, IEEE Transactions on*, *22*(6), 456–462.

[40] **White, S.** (1989). Applications of Distributed Arithmetic to Digital Signal Processing: A Tutorial Review, *ASSP Magazine, IEEE*, *6*(3), 4–19.

[41] **Martina, M.**, **Masera, G.**, **Roch, M.R. and Piccinini, G.** (2015). Result-Biased Distributed-Arithmetic-Based Filter Architectures for Approximately Computing the DWT, *IEEE Transactions on Circuits and Systems I: Regular Papers*, *62*(8), 2103–2113.

[42] **Alzaq, H.Y. and Ustundag, B.B.** (2017). Very-low-SNR cognitive receiver based on wavelet preprocessed signal patterns and neural network, *EURASIP Journal on Wireless Communications and Networking*, *2017*(1), 120, https://doi.org/10.1186/s13638-017-0902-7.

[43] **Carta, N.**, **Pani, D. and Raffo, L.** (2015). Impact of Threshold Computation Methods in Hardware Wavelet Denoising Implementations for Neural Signal Processing, G. Plantier, T. Schultz, A. Fred and H. Gamboa, editors, Biomedical Engineering Systems and Technologies: 7th International Joint Conference, BIOSTEC, volume 511, Springer International Publishing, Cham, Switzerland, pp.66–81, http://dx.doi.org/10.1007/978-3-319-26129-4_5.

[44] **Vishwanath, M.** (1994). The Recursive Pyramid Algorithm for The Discrete Wavelet Transform, *IEEE Transactions on Signal Processing*, *42*(3), 673–676.

[45] **Vetterli, M. and Herley, C.** (1992). Wavelets and Filter Banks: Theory and Design, *Signal Processing, IEEE Transactions on*, *40*(9), 2207–2232.

[46] **Chen, S.W. and Chen, Y.H.** (2015). Hardware Design and Implementation of a Wavelet De-Noising Procedure for Medical Signal Preprocessing, *Sensors*, *15*(10), 26396–26414, http://www.mdpi.com/1424-8220/15/10/26396.

[47] **Duan, F.**, **Dai, L.**, **Chang, W.**, **Chen, Z.**, **Zhu, C. and Li, W.** (2016). sEMG-Based Identification of Hand Motion Commands Using Wavelet Neural Network Combined With Discrete Wavelet Transform, *IEEE Transactions on Industrial Electronics*, *63*(3), 1923–1934.

[48] **Daubechies, I. and Sweldens, W.** (1998). Factoring Wavelet Transforms into Lifting Steps, *Journal of Fourier Analysis and Applications*, *4*(3), 247–269, http://dx.doi.org/10.1007/BF02476026.

[49] **Meher, P.K.**, **Mohanty, B.K. and Swamy, M.M.S.** (2015). Low-Area and Low-Power Reconfigurable Architecture for Convolution-Based 1-D DWT Using 9/7 and 5/3 Filters, 2015 28th International Conference on VLSI Design, pp.327–332.

[50] **Madanayake, A.**, **Cintra, R.J.**, **Dimitrov, V.**, **Bayer, F.**, **Wahid, K.A.**, **Kulasekera, S.**, **Edirisuriya, A.**, **Potluri, U.**, **Madishetty, S. and Rajapaksha, N.** (2015). Low-Power VLSI Architectures for DCT/DWT: Precision vs Approximation for HD Video, Biomedical, and Smart Antenna Applications, *IEEE Circuits and Systems Magazine*, *15*(1), 25–47.

[51] **Taylor, F.** (1984). Residue Arithmetic A Tutorial with Examples, *Computer*, *17*(5), 50–62.

[52] **Pontarelli, S.**, **Cardarilli, G.**, **Re, M. and Salsano, A.** (2012). Optimized Implementation of RNS FIR Filters Based on FPGAs, *Journal of Signal Processing Systems*, *67*(3), 201–212, http://dx.doi.org/10.1007/s11265-010-0537-y.

[53] **Rosen, K.H.** (2004). *Elementary Number Theory and its Applications*, Addison-Wesley, Reading, MA, 5th edition.

[54] **Cardarilli, G.C.**, **Nannarelli, A.**, **Petricca, M. and Re, M.** (2015). Characterization of RNS Multiply-Add Units for Power Efficient DSP, 2015 IEEE 58th International Midwest Symposium on Circuits and Systems (MWSCAS), pp.1–4.

[55] **Reddy, K.**, **Bajaj, S. and Kumar, S.** (2014). Shift Add Approach Based Implementation of RNS-FIR Filter Using Modified Product Encoder, TENCON 2014 - 2014 IEEE Region 10 Conference, pp.1–6.

[56] **Hariri, A.**, **Navi, K. and Rastegar, R.** (2008). A New High Dynamic Range Moduli Set with Efficient Reverse Converter, *Computers & Mathematics with Applications*, *55*(4), 660 – 668, http://www.sciencedirect.com/science/article/pii/S0898122107004993.

[57] **Lin, S.H.**, **hwa Sheu, M.**, **Wang, C.H. and Kuo, Y.C.** (2008). Area-Time-Power Efficient VLSI Design for Residue-to-binary Converter Based on Moduli Set $(2^n, 2^{n+1}-1, 2^n+1)$, Circuits and Systems, 2008. APCCAS 2008. IEEE Asia Pacific Conference on, pp.168–171.

[58] **Jarrah, A. and Jamali, M.M.** (2014). Optimized FPGA based implementation of discrete wavelet transform, 2014 48th Asilomar Conference on Signals, Systems and Computers, pp.1839–1842.

[59] **I, M.**, **Tripathi, S. and TSB, S.** (2016). Pipelined architecture for filter bank based 1-D DWT, 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN), pp.47–52.

[60] **Madishetty, S.K.**, **Madanayake, A.**, **Cintra, R.J. and Dimitrov, V.S.** (2014). Precise VLSI Architecture for AI Based 1-D/ 2-D Daub-6 Wavelet Filter Banks With Low Adder-Count, *IEEE Transactions on Circuits and Systems I: Regular Papers*, *61*(7), 1984–1993.

[61] **Avinash, C.S. and Alex, J.S.R.** (2015). FPGA implementation of Discrete Wavelet Transform using Distributed Arithmetic Architecture, 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), pp.326–330.

[62] **Yang, P. and Li, Q.** (2014). Wavelet Transform-Based Feature Extraction for Ultrasonic Flaw Signal Classification, *Neural Computing and Applications*, *24*(3-4), 817–826, `http://dx.doi.org/10.1007/s00521-012-1305-7`.

[63] **Gnavi, S.**, **Penna, B.**, **Grangetto, M.**, **Magli, E. and Olmo, G.** (2002). Wavelet Kernels on a DSP: A Comparison between Lifting and Filter Banks for Image Coding, *EURASIP Journal on Advances in Signal Processing*, *2002*(9), 458215, `https://doi.org/10.1155/S1110865702204126`.

[64] **M., M.A.B. and Sk, N.M.** (2018). An Efficient VLSI Architecture for Convolution Based DWT Using MAC, 2018 31st International Conference on VLSI Design and 2018 17th International Conference on Embedded Systems (VLSID), pp.271–276.

[65] **Gacic, A.**, **Puschel, M. and Moura, J.M.F.** (2004). Automatically generated high-performance code for discrete wavelet transforms, 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, volume 5, pp.V–69–72 vol.5.

[66] **Ramola, E. and Manoharan, J.S.** (2011). An area efficient VLSI realization of Discrete Wavelet Transform for multiresolution analysis, 2011 3rd International Conference on Electronics Computer Technology, volume 6, pp.377–381.

[67] **Mamatha, I.**, **Tripathi, S. and Sudarshan, T.S.B.** (2017). Convolution based efficient architecture for 1-D DWT, 2017 International Conference on Computing, Communication and Automation (ICCCA), pp.1436–1440.

[68] **Ramírez, J.**, **García, A.**, **Fernandez, P.G. and Lloris, A.** (2000). An efficient RNS architecture for the computation of discrete wavelet transforms on programmable devices, 2000 10th European Signal Processing Conference, pp.1–4.

[69] **Aksoy, L.**, **Flores, P. and Monteiro, J.** (2014). A Tutorial on Multiplierless Design of FIR Filters: Algorithms and Architectures, *Circuits, Systems, and Signal Processing*, *33*(6), 1689–1719, `https://doi.org/10.1007/s00034-013-9727-8`.

[70] **Voronenko, Y. and Püschel, M.** (2007). Multiplierless Multiple Constant Multiplication, *ACM Trans. Algorithms*, *3*(2), `http://doi.acm.org/10.1145/1240233.1240234`.

[71] **Al-Hasani, F.**, **Hayes, M.P. and Bainbridge-Smith, A.** (2013). A Common Subexpression Elimination Tree Algorithm, *IEEE Transactions on Circuits and Systems I: Regular Papers*, *60*(9), 2389–2400.

[72] **Lou, X.**, **Yu, Y.J. and Meher, P.K.** (2015). New Approach to the Reduction of Sign-Extension Overhead for Efficient Implementation of Multiple Constant Multiplications, *IEEE Transactions on Circuits and Systems I: Regular Papers*, *62*(11), 2695–2705.

[73] **Liu, H. and Jiang, A.** (2016). Efficient design of FIR filters using common subexpression elimination, 2016 8th International Conference on Wireless Communications Signal Processing (WCSP), pp.1–5.

[74] **Aksoy, L.**, **Flores, P. and Monteiro, J.** (2014). Multiplierless Design of Folded DSP Blocks, *ACM Trans. Des. Autom. Electron. Syst.*, *20*(1), 14:1–14:24, `http://doi.acm.org/10.1145/2663343`.

[75] **Allred, D.J.**, **Huang, W.**, **Krishnan, V.**, **Yoo, H. and Anderson, D.V.** (2004). An FPGA Implementation for a High Throughput Adaptive Filter using Distributed Arithmetic, Field-Programmable Custom Computing Machines, 2004. FCCM 2004. 12th Annual IEEE Symposium on, pp.324–325.

[76] **Yoo, H. and Anderson, D.V.** (2005). Hardware-efficient Distributed Arithmetic Architecture for High-Order Digital Filters, Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005., volume 5, pp.v/125–v/128 Vol. 5.

[77] **Jenkins, W. and Leon, B.** (1977). The use of Residue Number Systems in the Design of Finite Impulse Response Digital Filters, *Circuits and Systems, IEEE Transactions on*, *24*(4), 191–201.

[78] **Chang, C.H.**, **Molahosseini, A.S.**, **Zarandi, A.A.E. and Tay, T.F.** (2015). Residue Number Systems: A New Paradigm to Datapath Optimization for Low-Power and High-Performance Digital Signal Processing Applications, *IEEE Circuits and Systems Magazine*, *15*(4), 26–44.

[79] **Ramírez, J.**, **Meyer-Bäse, U.**, **Taylor, F.**, **García, A. and Lloris, A.** (2003). Design and Implementation of High-Performance RNS Wavelet Processors Using Custom IC Technologies, *Journal of VLSI signal processing systems for signal, image and video technology*, *34*(3), 227–237, http://dx.doi.org/10.1023/A:1023296218588.

[80] **Conway, R. and Nelson, J.** (2004). Improved RNS FIR Filter Architectures, *IEEE Transactions on Circuits and Systems II: Express Briefs*, *51*(1), 26–28.

[81] **Mohan, P.** (2007). RNS-To-Binary Converter for a New Three-Moduli Set $2^{n+1}-1, 2^n, 2^n-1$, *Circuits and Systems II: Express Briefs, IEEE Transactions on*, *54*(9), 775–779.

[82] **Vun, C.H.**, **Premkumar, A. and Zhang, W.** (2013). A New RNS based DA Approach for Inner Product Computation, *Circuits and Systems I: Regular Papers, IEEE Transactions on*, *60*(8), 2139–2152.

[83] **Cao, B.**, **Srikanthan, T. and Chang, C.H.** (2005). Efficient Reverse Converters for the Four-moduli sets $(2^n-1, 2^n, 2^n+1, 2^{n+1}-1)$ and $(2^n-1, 2^n, 2^n+1, 2^{n-1}-1)$, *IEE Proc Comput Digit Tec*, *152*(5), 687–696.

[84] **Cao, B.**, **Chang, C.H. and Srikanthan, T.** (2003). An Efficient Reverse Converter for the 4-moduli set $2^n-1, 2^n, 2^n+1, 2^{2n}+1$ Based on the new Chinese Remainder Theorem, *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, *50*(10), 1296–1303.

[85] **Zimmermann, R.** (1999). Efficient VLSI implementation of modulo (2n plusmn;1) addition and multiplication, Proceedings 14th IEEE Symposium on Computer Arithmetic (Cat. No.99CB36336), pp.158–167.

[86] **Kalampoukas, L.**, **Nikolos, D.**, **Efstathiou, C.**, **Vergos, H.T. and Kalamatianos, J.** (2000). High-Speed Parallel-Prefix Modulo 2^n-1 Adders, *IEEE Transactions on Computers*, *49*(7), 673–680, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=863036.

[87] **Dimitrakopoulos, G.**, **Nikolos, D.G.**, **Vergos, H.T.**, **Nikolos, D. and Efstathiou, C.** (2005). New architectures for modulo 2^n-1 adders, 2005 12th IEEE International Conference on Electronics, Circuits and Systems, pp.1–4.

[88] Xilinx Inc., Zynq-7000 All Programmable SoC ZC706 Evaluation Kit, https://www.xilinx.com/products/boards-and-kits/ek-z7-zc706-g.html, accessed: 2017-08-30.

[89] **Xilinx**, (2011). LogiCORE IP FIR Compiler v6.3, http://www.xilinx.com/support/documentation/ip_documentation/fir_compiler/v6_3/ds795_fir_compiler.pdf.

[90] **wireless Innovation Forum**, 4DSP - FMC30RF, available at http://www.wirelessinnovation.org.

[91] **Qadir, J.** (2013). Artificial Intelligence Based Cognitive Routing for Cognitive Radio Networks, *CoRR*, *abs/1309.0085*, http://arxiv.org/abs/1309.0085, 1309.0085.

[92] **Salih, Q.M.**, **Rahman, M.A.**, **Bhuiyan, M.Z.A. and Azmi, Z.R.M.** (2017). The Art of Using Cross-Layer Design in Cognitive Radio Networks, G. Wang, M. Atiquzzaman, Z. Yan and K.K.R. Choo, editors, Security, Privacy, and Anonymity in Computation, Communication, and Storage, Springer International Publishing, Cham, pp.544–556.

[93] **Liu, Q.**, **Zhou, S. and Giannakis, G.** (2004). Cross-Layer combining of adaptive Modulation and coding with truncated ARQ over wireless links, *IEEE Transactions on Wireless Communications*, *3*(5), 1746–1755.

[94] **Alzaq, H.Y. and Ustundag, B.B.** (2018). An optimized two-level discrete wavelet implementation using residue number system, *EURASIP Journal on Advances in Signal Processing*, *2018*(1), 41, https://doi.org/10.1186/s13634-018-0559-3.

[95] UG-570: AD9361 Reference Manual (Rev. A), http://www.analog.com/media/en/technical-documentation/user-guides/AD9361_Reference_Manual_UG-570.pdf.

[96] Xilinx Inc., System Generator for DSP, available at http://www.xilinx.com/products/design-tools/vivado/integration/sysgen.html.

[97] Vivado Design Suite, https://www.xilinx.com/products/design-tools/vivado.html.

CURRICULUM VITAE

Name Surname: Husam Y. I. ALZAQ

Place and Date of Birth: United Arab Emirates, 27 June, 1979

E-Mail: alzaq@itu.edu.tr, h.y.alzaq@gmail.com

EDUCATION:

- **B.Sc.:** 2002, Islamic University of Gaza, Computer Engineering Department, Palestine.
- **M.Sc.:** 2007, Computational Science in Engineering, Technical University of Braunschweig, Braunschweig, Germany.

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2007-2010 Lecturer, Computer Engineering Department, IUGAZA, Gaza, Palestine.
- 2005-2007 Research Assistant, Institute of surface technology, TU-Braunschweig, Germany.
- 2019 Completed Doctorate at Istanbul Technical University

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Alzaq H.**, and Ustundag, B.B., 2018. A Comparative Analysis of 1-Level Multiplier-free Discrete Wavelet Transform Implementations on FPGAs, *Turkish Journal of Electrical Engineering & Computer Science*, 26(5), 1190-2004. doi:10.3906/elk-1707-101
- **Alzaq H. Y.**, Ustundag B. B., 2017. Very-low-SNR cognitive receiver based on wavelet preprocessed signal patterns and neural network, *EURASIP Journal on Wireless Communications and Networking,* 2017(1), 120. doi:10.1186/s13638-017-0902-7
- **Alzaq H.Y.**, Ustundag, B.B., 2018. An Optimized 2-Level Discrete Wavelet Implementation Using Residue Number System. *EURASIP Journal on Advances in Signal Processing,* 2018(1), 41. doi:10.1186/s13634-018-0559-3
- **Alzaq H.**, Ustundag B. B., 2017. Multiplier-less 1-Level Discrete Wavelet Transform Implementations on ZC706 development kit. *10th International Conference on Electrical and Electronics Engineering, ELECO 2017*, Nov 31- Dec 2, 2017, Bursa, Turkey.

- **Alzaq H.**, Ustundag B. B., 2017. A comparative analysis of multiplier-less 1-level discrete wavelet transform implementations on FPGAs. *2017 International Conference on Engineering and Technology (ICET)*, Aug 21-23, 2017, Antalya, Turkey. doi: 10.1109/ICEngTechnol.2017.8308194.

- **Alzaq H.**, Ustundag B. B., 2015. Wavelet Preprocessed Neural Network Based Receiver for Low SNR Communication System, *Proceedings of European Wireless 2015; 21th European Wireless Conference*, May 21-24, 2015 Budapest, Hungary.

- **Husam Alzaq** and Burak Berk Üstündağ, 2018. A Comparative Performance of Discrete Wavelet Transform Implementations Using Multiplierless, *Wavelet Theory and Its Applications* Editor Sudhakar Radhakrishnan, IntechOpen, DOI:10.5772/intechopen.76522.

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- **Alzaq H.**, Kabadayi S., 2013. Mobile robot comes to the rescue in a WSN. *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1977-1982, London, Grate Britain. doi: 10.1109/PIMRC.2013.6666468