Design and Implementation of a Dynamic Partial Reconfigurable Demodulation System for Satellite Receivers

Kyushu Institute of Technology

Department of Engineering

Graduate School of Engineering

Design and Implementation of a Dynamic Partial Reconfigurable Demodulation System for Satellite Receivers

By

Mohamed Elhady Magdy Mohamed Gad Keshk

A DISSERTATION SUBMITTED TO THE
GRADUATE SCHOOL OF ENGINEERING
AT KYUSHU INSTITUTE OF TECHNOLOGY
IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN APPLIED SCIENCE FOR INTEGRATED SYSTEM
ENGINEERRING

Graduate School of Engineering, Kyushu Institute of Technology
Kitakyushu, Japan
2018

Kyushu Institute of Technology

Department of Engineering

Graduate School of Engineering

Design and Implementation of a Dynamic Partial Reconfigurable Demodulation System for Satellite Receivers

By

Mohamed Elhady Magdy Mohamed Gad Keshk

A DISSERTATION SUBMITTED TO THE
GRADUATE SCHOOL OF ENGINEERING
AT KYUSHU INSTITUTE OF TECHNOLOGY
IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN APPLIED SCIENCE FOR INTEGRATED SYSTEM
ENGINEERRING

Dissertation Supervisor

Prof. Kenichi Asami

Dept. of Applied Science for Integrated System Engineering
Kyushu Institute of Technology

Kyushu Institute of Technology

Department of Engineering

Graduate School of Engineering

Design and Implementation of a Dynamic Partial Reconfigurable Demodulation System for Satellite Receivers

By

Mohamed Elhady Magdy Mohamed Gad Keshk

A DISSERTATION SUBMITTED TO THE
GRADUATE SCHOOL OF ENGINEERING
AT KYUSHU INSTITUTE OF TECHNOLOGY
IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN APPLIED SCIENCE FOR INTEGRATED SYSTEM
ENGINEERRING

Approved by Examiners Committee:

Prof. KENICHI ASAMI Kenichi Asami

Prof. MENGU CHO Cho Mengu

Prof. KEI-ICHI OKUYAMA Keiichi Okuyama

Prof. KAZUHIRO TOYODA Kazuhiro Toyoda

Prof. MOTOKI MIURA Motoki Miura

بسم الله الرحمن الرحيم

وَمَا أُوتِيتُم مِّنَ الْعِلْمِ إِلَّا قَلِيلًا

صدق الله العظيم

“Success is not final, failure is not fatal:

It is the courage to continue that counts.”

By

- Winston S. Churchill -

DEDICATION

To the soul of My Father, My great Mother,

My pretty Spouse,

And

My son

Without their support, this labor would not have been possible

AKNOWLEDGMENTS

First and foremost, all praises and gratitude means reciprocating gentleness to the one who has done worthy and kind things to me. The one who is most commendable of thanks and praise is Allah, the almighty, because of the great favoritisms and blessings that he has bestowed upon his slaves in both spiritual and worldly terms and who gave me the ability and bestowed me with perseverance to write this thesis.

I would like to express my sincere appreciation and thanks to my supervisor Professor KENICHI ASAMI, you have been a tremendous mentor for me. I would like to thank you for encouraging my research and for allowing me to grow as a research scientist. Your advice on both research as well as on my career have been priceless. I would also like to thank my committee members, Professor CHO MENGU, Professor KEI-ICHI OKUYAMA, Professor KAZUHIRO TOYODA, and Professor MOTOKI MIURA for serving as my committee members even at hardship. I also want to thank you for letting my mid-term evaluation and final defense be an enjoyable moment, and for your brilliant comments and suggestions, thanks so much to you. I would especially like to thank all members: Asami Lab members (Doctors, Masters, and Bachelors), Ten-koh satellite project and Cho Lab members (Post Docs, Doctors, Masters, and Bachelors) in the department of applied sciences and integrated system Engineering of Kyushu Institute of Technology (Kitakyushu, Japan). All of you have been there to support and help me to accomplish my doctoral study when I started my research.

A Lot of thanks especially Dr. Werner Balogh to the UNOOSA United Nations Office Outer Space Affairs in Austria, which offers me Japan Long-Term Fellowship Program 2015 till 2018 for Post-graduate study on Nano-Satellite Technologies (PNST) in collaboration with Egyptian Space Program and Kyushu Institute of Technology.

A special thanks to my family. Words cannot express how grateful I am to my parents, brother, and Sister, for all of the sacrifices that you've made on my behalf. Your prayer to

Allah for me was what sustained me thus far. I would also like to thank all of my Muslim friends (Professor Yasser, Dr. Yahia, Dr. Sidi, Emad, Ibrahim, Gaber, Hisham, Tamer, Rahmi, Fathin, …) who respondents for the support, willingness and enchanted me to strive towards my goal.

The participants of Asami laboratory (Amartuvshin, Rafael, Cosmas, Tuguldur, and all the others) and many other students from other laboratories (Dr. Dima, Bonsu, Apiwat, Fukuda, ...) have been great relations and colleagues and I am appreciative for the great time we spent together, for our wonderful trips and for all the great retentions.

I would like express appreciation to my beloved wife and my baby who spent sleepless nights with and were always my support in the moments when there was no one to answer my queries.

PREFACE

The collaboration among countries in space field promotes the financial efficiency and political sustainability. The space faring nations have most of the responsibility to enhance and make the best use of such collaborations. There are many collaborations aim. The high cost of building a satellite and its ground infrastructure causes the increment of the number of international cooperation in space field to support and encourage the developing and emerging countries to have a position in space and participate in space activities by developing small or lean satellites without affording huge costs.

These collaborations are aimed to build the human capacity of these countries by participating in joint satellites projects. Instead of developing ground infrastructure which needs huge cost and takes a lot of time, these countries can use the existing facilities of the faring countries. The existing ground stations may face the problem of using different modulation and demodulation techniques limits the number of these stations that can communicate with the satellites. Developing the ground stations and/or satellites by installing several transceivers can be used as an alternative to overcome this problem but this solution increases both the complexity, power consumption, and the cost of both ground station and satellite. The same problem faces the communication between a satellite and a relay satellite.

This research aims to overcome such a problem by implementing a prototype of a dynamic partial reconfigurable demodulation system that can be used as a core of a generic receiver for the satellite receivers. This system performs the role of a generic receiver but with low complexity, power consumption, and cost of a satellite. In fact, the system is not only promoting the international collaboration, but also it overcomes the channel problems by alternating the modulation and demodulation techniques based on channel condition, which enhances the communication quality. The system is based on pattern recognition approach, software defined radio and FPGA dynamic partial

reconfiguration technology to reduce the hardware complexity and obsolescence, power consumption, and cost of a satellite.

The dynamic partial reconfigurable demodulator system is divided into 2 main sections; automatic digital modulation recognition (ADMR), and dynamic partial reconfiguration (DPR) sections. The ADMR can be used in relay satellites and other satellites where the demodulation and re-modulation of the data is performed onboard. In addition, ADMR can be used to overcome channel problems during the communication session with a satellite as bandwidth congestion/crunch by changing the existing modulation scheme by another more efficient bandwidth modulation scheme. ADMR can be used to secure the received and transmitted signal by locating, classifying and preventing the unwanted signals. ADMR can be used in interference identification, spectrum monitoring, signal confirmation, electronic warfare, cognitive radio, threat analysis, and signal surveillance.

ADMR is responsible of the classification of the modulation type of the received modulated signal and it consists of features extraction and SVM classifier modules. Features module is to extract the features of the received modulated signal which in this thesis is the discreet wavelet transform (DWT) coefficients level3. The realization of this module on FPGA requires the use 3 DWT stages each stage is implemented by a finite impulse response filter. Two low pass filters and one high pass filter is required for the implementation. These features are the input to the SVM classifier.

SVM classifier is a binary classifier that uses to classify two received modulation types BPSK and QPSK. The classifier has two stages; training stage which is performed using MATLAB to train the classifier with the features of both BPSK and QPSK modulations. The aim of this stage is to construct the classifier then the resulting parameters will be used to design and implement the classifier on FPGA. The second stage is the classification stage which classify the received modulation types based on the prior generated parameters (weights, Alpha, bias) during training stage and using radial base function as a SVM kernel function. The classification result is the input of the DPR section.

The DPR section consists of partial reconfiguration controller (PRC), external memory controller (EMC) and internal configuration access port (ICAP). PRC is the controller of the whole reconfiguration process. It stores the addresses of the BPSK and QPSK demodulator on external flash memory and according to the classification result, which act as a trigger to DPR controller, DPR sends the corresponding demodulator address to EMC to fetch the demodulator partial bitstream from external memory then resends this bitstream file to DPR controller. Then, DPR controller sends the partial bitstream file to ICAP port to reconfigure the reconfigurable partition by the proper demodulator. After that, the original binary sequence is obtained.

The BPSK and QPSK demodulators are implemented based on SDR using a sine and cosine lookup table instead of sine and cosine wave generator. The demodulation is done by multiplying the samples of the received signal by the sine/cosine samples that stored into the lookup tables.

The performance of the system is evaluated by implementing a BPSK and QPSK modulator on a kintex-7 FPGA board to act as the received signal for dynamic partial demodulator system that implemented into another kintex-7 board to compute the classification rate with the presence of AWGN with SNR from -10 to 20 dB, classification time, and reconfiguration time.

In this dissertation, the multiple access of ICAP primitive by several application is approved. The DPRDS and Soft Error Mitigation (SEM) systems are used to run together on FPGA and acquired the access of ICAP primitive. The results showed that the implementation of DPR design will not prevent the SEM application from the access of the ICAP primitive and vice versa.

Using the Xilinx essential bit technology reduces the time required to inject and mitigate the errors in the DPRDS design by a 96%. The main reason is essential bit technology based of injecting the error only into the bits of the configuration frame that are used by the design and ignoring the other bits. To get the essential bits, Essential Bits Configuration (EBC) and Essential Bits Definition (EBD) files have to be generated for comparing the bits existing into EBD file with the corresponding bits of EBC file, where

bits of value 1 in EBD file mean the corresponding bits in EBC file are essential, otherwise, the corresponding bits in EBC file are not essential.

A MATLAB script is used to calculate the Logical Addresses (LAs) of these bits, and control and monitor the injection and mitigation of the errors on FPGA, through UART interface, in both static and dynamic sections of the DPRDS design on the conditions of SEM IP core and the results showed that the both DPRDS and SEM applications are working properly.

Table of Contents

List of Figures

List of Tables

List of Symbols

Symbol	Description
$s_{SPK}(t)$	BPSK modulated Signal in time domain
f_c	Carrier frequency
T_b	Bit duration
BW	Bandwidth
$C(t)$	Carrier signal in time domain
$s_{QSPK}(t)$	BPSK modulated Signal in time domain
W	Weight vector
X	Input features vector
b	Bias, scaler quantity
$f(x)$	Classification equation
y_i	Class label
x_i	Support vectors vector
C	Penalty value
α_i	Lagrangian multiplier
ε	Marginal error
f_{out}	NCO output frequency
σ^2	variance

List of Acronyms

Abbreviation	Description
ADC	Analog to Digital Converter
ADMC	Automatic Digital Modulation Classification
ADMR	Automatic Digital Modulation Recognition
ALU	Arithmetic Logic Units
AM	Amplitude Modulation
AMR	Automatic Modulation Recognition
ANN	Artificial Neural Network ANN
ASIC	Application Specific Integrated Circuits
AWGN	Additive White Gaussian Noise
BASK	Binary Amplitude Shift Keying
BFSK	Binary Frequency Shift Keying
BPI	Byte Peripheral Interface
BPSK	Binary Phase Shift Keying
BSTI	Basic Space Technology Initiative
CC	Cyclic Cumulants
CLB	Configurable Logic Blocks
COPUOS	Committee on the Peaceful Uses of Outer Space
COTS	Commercial Off-the Shelf
CPP	Configuration Packet Processor
CRC	Cyclic Redundancy Check
CT	Configuration Time

DAC	Digital to analog Converter
DCT	Discrete Cosine Transform
DDC	Digital Downconverter
DNST	Doctorate in Nano-Satellite Technologies
DPR	Dynamic Partial Reconfiguration
DPRDS	Dynamic Partial Reconfigurable Demodulators System
DSM	Digital Signal Modulator
DSP	Digital Signal Processor
DST	Discrete Sine Transform
DT	Decision-Theoretic
DUC	Digital Upconverter
DUT	Device Under Test
DWT	Discrete Wavelet Transform
ECC	Frame Error Correcting Code
EMC	External Memory Controller
ESA	European Space Agency
ESP	Egyptian Space Program
FCC	Federal Communications Commission
FDR	Frame Data Register
FFs	Flip Flops
FIR	Finite Impulse Response
FIT	Failures in Time
FM	Frequency Modulation

FPGA	Field Programmable Gate Array
FSK	Frequency Shift Keying
FSM	Finite State Machine
GCSs	Ground Control Stations
GMSK	Gaussian Minimum Shift Keying
GPP	General Purpose Processor
HOS	Higher Order Statistics
HPF	High Pass Filter
I/O Blocks	Input/ Output Blocks
ICAP	Internal Configuration Access Port
IF	Intermediate Frequency
ILA	Integrated Logic Analyzer
IRSO	Indian Research Space Organization
ISS	launching International space Station
JTAG	Joint Test Action Group
k-NN	k-Nearest Neighbor
KUAR	GNU Radio, Kansas University Agile Radio
LA	Logical Address
LDA	Linear Discriminant Analysis
LPF	Low Pass Filter
LUTs	Lookup Tables
MBU	Multiple Bit Upset
MFSK	M-ary Frequency Shift Keying

MIMO	Multiple Input Multiple Output
MLPNN	Multi-Layer Perceptron Neural Network
MMCM	a Mixed Mode Clock Manager
MSK	M-ary Phase Shift Keying MPSK
NARSS	National Authority of Remote sensing and Space Sciences
NCD	Native Circuit Description
NCO	Numerical Controlled Oscillator
NICT	The Japanese National Institute of Information and Communications Technology
NN	Naive Bayes (NB)
OFDM	Orthogonal Frequency Division Multiplexing
PAPR	Peak to Average Power ratio
PLL	Phase Locked Loop
PNST	Post-graduate study of Nano-Satellite Technologies
PR	Pattern Recognition
PR	Partial Reconfiguration
PRC	Partial Reconfiguration Controller
PRM	Partial Reconfigurable Module
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RBF	Radial Base Function
RF	Radio Frequency

RM	Reconfigurable Module
SAARC	South Asian Association for Regional Cooperation
SDR	Software Defined Radio
SEIC	Space Engineering International Courses
SEM	Soft Error Mitigation
SEU	Single Event Upset
SNR	Signal to Noise Ration
SOC	System-On-Chip
SPENVIS	Space Environment Information System
SPI	Serial Peripheral Interface
S-RAM	Static Random Access Memory
SVM	Support Vector Machine
SVs	Support Vectors
UNOOSA	United Nations Office for Outer Space Affairs
USRP	Universal Software Radio Peripheral
WLAN	Wireless Local Area Network

Chapter 1 : Introduction

Recently, the number of international cooperation in space field increases rapidly. Several benefits can be gained from international cooperation such as funding, political sustainability, and financial efficiency. International cooperation has several topics and goals. For example, cooperation in space science, earth observation, deep space, interplanetary exploration, capacity building and technology transfer. After the end of cold war by many years, the first cooperation in space field is started by developing and launching International Space Station (ISS) in 1998 and involving U.S., Soviet union, Japan, Canada, and European Space Agency (ESA). Many experiments can be performed in space such as life sciences, space medicine, physical sciences where the conditions are achieved regarding the difficulties face these experiments to be performed on the earth which enhances the human science and promote the technological progress [1].

In twenty-first century, the regulation of international cooperation in Space exploration activities would be needed, so there was a necessary to set a theoretical framework that is able to manage all international cooperation. This framework explains all aspects of the cooperation and contains models of cooperation to illustrate how actors and initial conditions (political, scientific, economic, and technological) can work together to achieve promising outcomes [2]. United Nations Office for Outer Space Affairs (UNOOSA), Committee on the Peaceful Uses of Outer Space (COPUOS) in 1996 adopted a declaration to organize such type of international cooperation.

Indian Research Space Organization (IRSO) has rapid development in space science as the result of its activities in both economic and social fields based on its outer space exploration activities. IRSO engaged international cooperation with South Asian Association for Regional Cooperation (SAARC) in SAARC satellite project to promote the socio-economic development in that area, which improves and promotes the economy and security of India [3]. National Authority of Remote sensing and Space Sciences

(NARSS) representative by Egyptian Space Program (ESP) had engaged with Ukraine in Egypt-Sat1 satellite project which considered the first Egyptian remote sensing satellite. This cooperation encompassed all the processes of designing, manufacturing, and testing the satellite, in addition, technology transfer and capacity building.

Egypt is not the only North African country that engaged in international cooperation, and other countries such as Algeria and Morocco have international cooperation too. Morocco as an example, has many cooperation with European Union, ESA and other space agencies especially in micro-satellite projects for remote sensing purposes [4], so to get the most benefits of international cooperation, it is required to overcome the problems that can obstacle the cooperation. There are many trails performed among Latin America countries to go into such cooperation by overcome the regional problems to establish regional cooperation mechanism in space [5]. The same situation with India-Canada cooperation [6], and Canada-China cooperation [7] to overcome any obstacles prevents the fruitful outcomes of the cooperation.

International cooperation has several advantages and can be applied in several fields. Sharing Ground Control Stations (GCSs), capacity building, building satellites, establishment of facilities required for satellites development, space applications, and education can be a form of international cooperation. Basic Space Technology Initiative (BSTI) created by UNOSSA is aiming to spread the space engineering technology among the developing and emerging countries to start their first step toward space technology. BSTI and Kyushu Institute of Technology has a long time collaboration in space education programs under the framework of United Nation/Japan Long-Term Fellowship Program starting from Doctorate in Nano-Satellite Technologies (DNST), 2010, passing through Post-graduate study of Nano-Satellite Technologies (PNST), 2013-2018 which gives the chance for up to three students for Master course and up to three students for the Doctor course to study in Space Engineering International Courses (SEIC). The PNST program has been extended to 2020 to give the students from all over the world new opportunity to join the program.

I have got the opportunity to join the PNST program at Kyushu Institute of Technology and be one of the Ph.D. students that attended the SEIC course and used the Testing facilities of the institute. Of course, this is certainly considered elegant advantage and fruitful outcome of the international cooperation.

This research point promotes the international cooperation in both ground and space segments by implementing a prototype of Dynamic Partial Reconfigurable Demodulation System (DPRDS) that can be used as a main core of a generic receiver. Generic receiver can be implemented in different categories of the satellites to increase the number of Ground Control Stations (GCSs) where a satellite can communicate without the need to install several receivers where increase the complexity and the cost of a satellite. Generic receiver can promote the collaboration among countries in several fields especially remote sensing. This prototype would acquire the benefits of the rapid development in microelectronic technologies especially in FPGA. The prototype is implemented on FPGA which is based on the concept of Software Defined Radio (SDR) receivers to implement several demodulators defined in software, and FPGA partial reconfiguration technology that allows dynamic reconfigurability of a part of the design during runtime without any effects on the rest of the design.

Making the best use of the existing technology and the rapid development of these technologies provides suitable solutions to overcome some obstacles. The concept of SDR receivers states that some or all of hardware components of a radio communication can be implemented in software so as to be called software defined. This principle opens a new thinking field and smart way to reduce the complexity and cost and at the same completely finished the problem of obsoleting hardware. Simply you can change your design by replacing the old software with new one without the need of replacing costly hardware [8].

Rapid development in FPGA aims to increase speed of signal processing and exploration of techniques that can help in solving some challenges, which was in past is impossible to

be implemented. FPGA Dynamic Partial Reconfiguration (DPR) technique supports the designers to implement intelligent systems that were impossible to be implemented in real world. DPR capable of achieving space requirements starting from a high flexible System-On-Chip (SOC) to adaptive component algorithm [9]. DPR can be done by FPGA itself using Internal Configuration Access Port (ICAP) or by using external processor using Joint Test Action Group (JTAG). In the next sections, I would explain the motivation, objectives, and methodology of the research.

1.1. Problem Statement and Motivation

The high cost of building space infrastructure causes the increment of the number of international cooperation in space field and allows the chance to the developing and emerging countries to participate in space activities without affording huge costs and in other cases achieves political sustainability, and financial efficiency. So the removing of any obstacles that may prevent or reduce this cooperation is a challenge.

The main existing problem is the limited communication capability of the satellites whereas they can communicate only with GCSs that use the same modulation and demodulation schemes in their transceivers. For example, if the satellite use Binary Phase Shift Keying (BPSK) modulation/ demodulation technique, so it can demodulate the data of the received signal correctly if and only if the modulation of the received signal is BPSK too, otherwise, the demodulated data will not be correct as in Fig. 1-1. The same problem is applied for the communication between a satellite and a relay satellite. To solve this problem, satellites/GCSs receivers have to be more generic to permit communication with several relay satellites and GCSs that use different modulation and demodulation techniques without the need of using the traditional method that required installation of several receivers into satellites, therefore the complexity and cost of satellites increase.

This research aims to promote the international cooperation by using the recent development technologies by implementing a Dynamic Partial Reconfigurable

Demodulation System (DPRDS) where its design is based mainly on SDR principle and FPGA partial reconfiguration technique. DPRDS is the main core of a generic receiver and it considered a simple and low-cost method to implement a universal or generic receiver. DPRDS is considered the first data processing stage after receiver front end stage which contains antenna, low noise amplifier, and down conversion processes.

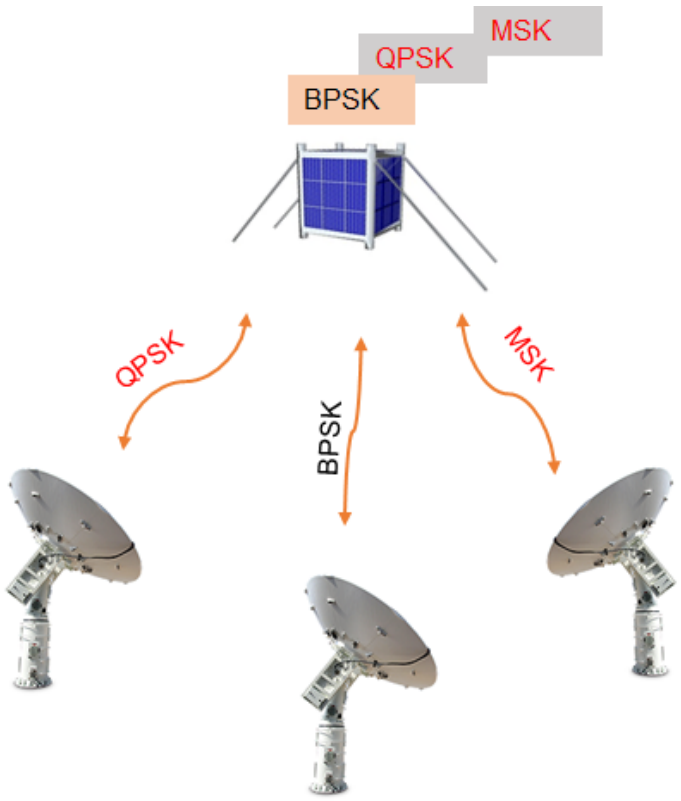


Figure 1-1. Problem Statement Configuration

The Automatic Digital Modulation Recognition (ADMR) or the classification part of the DPRDS system plays an important role between the detection of the signal and demodulation stages, which allows a satellite to adaptively demodulate the modulation scheme of the received modulated signal correctly after switching the modulation scheme by a GCS/s. There are many applications of ADMR in both civilian and military fields. The ADMR can be used in relay satellites and other satellites where the demodulation

and re-modulation of the data is performed onboard. In addition, ADMR can be used to overcome channel problems during the communication session with a satellite as bandwidth congestion/crunch by changing the existing modulation scheme by another more efficient bandwidth modulation scheme [35].

ADMR can be used to secure the received and transmitted signal by locating, classifying and preventing the unwanted signals. ADMR can be used in interference identification, spectrum monitoring, signal confirmation, electronic warfare, cognitive radio, threat analysis, and signal surveillance [23], [34].

National Aeronautics and Space Administration (NASA) is investigating cognitive radio technologies using ADMR for the improvement of the future communication architecture, which are expected to reduce the network operational complexity and interference to self and others, and increase science data return [88].

Motivations can be summarized as follows:

- Promotion of international cooperation in space field by developing generic GCSs/satellites receivers
    - Help developing and emerging countries to establish their space activities with low cost and less time by sharing existing facilities.
    - The countries own the facilities could promote political sustainability, and enhance their economic efficiency
- Investigation of the ability to implement a dynamic partial reconfigurable demodulation system prototype that can be used as a main core of a generic receiver.
- Implementation of BPSK and QPSK demodulators based on SDR principle to increase flexibility.
- Reduction of complexity and cost of both GCS and satellite.
- Reduced Hardware Obsolescence.

1.2. Research Objectives

This research has several objectives and can be summarized as follows:

- Development and implementation of dynamic partial reconfigurable demodulators system as the main core of a generic receiver using DPR technique, SDR principle, and classification approach.
- Investigation of the suitable classification approach, and tradeoff between classification approaches; Decision Theoretical and Pattern Recognition approaches.
- Investigation of the suitable features extractors and classifier for the design.
- Implementation of several demodulators by software instead of hardware complexity based on SDR principle.
- Investigate the suitable DPR methodology.
- Discussion of the ability of sharing ICAP primitives among DPR and the other user applications.
- Applying soft error mitigation in laboratory on DPRDS as a case study.

1.3. Research Originality

The originality of this research can be summarized as follows:

- Appling FPGA dynamic partial reconfiguration technique in communication subsystem which can be used in building satellites that can communicate with several ground-stations (by replacing a demodulator by another one on the fly), which increases cross support.
- Implementing a classifier before the dynamic partial reconfiguration stage allows a satellite to adaptively demodulate the modulation scheme of the received modulated signal correctly after changing the modulation scheme by a GCS.
- Increasing the capability of FPGA to implement several applications as dynamic partial reconfiguration, fault management system, and laboratory soft error

injection mitigation that requires the access to ICAP using a multiple access technique.

- Merging both MATLAB and VIVADO tools capabilities in order to reduce the resources utilization and low power consumption.

1.4. Research Methodology

- Study of classification approaches then selection of the suitable approach.
- Study of partial reconfiguration and dynamic partial reconfiguration and their development methods.
- Study of SDR principle and developing of demodulators based on SDR.
- Selection of the proper FPGA board to implement the design.
- Developing of the classification section of the design including features extraction and SVM classifier.
- Using MATLAB R2015b together with VIVADO 2015.4, to implement complex system and get coefficients, parameters, and other needed results instead of implementation of complex system on FPGA.
- Developing of DPRDS using DPR technique and SDR concept.
- Testing of the performance of DPRDS in the presence of Additive White Gaussian Noise (AWGN).
- Proving of the ability of DPRDS to be implemented with other user applications that need the access to the ICAP primitive.
- Developing of Single Event Upset (SEU) mitigation of DPRDS as a case study of the multiple access of ICAP primitive.
- Results dissemination.

1.5. Dissertation Challenges

In this dissertation several principles, approaches, and techniques are used to achieve low power, less resources utilization and complexity, and low cost DPRDS. So the dissertation has four challenges as illustrated in Fig. 1-2.

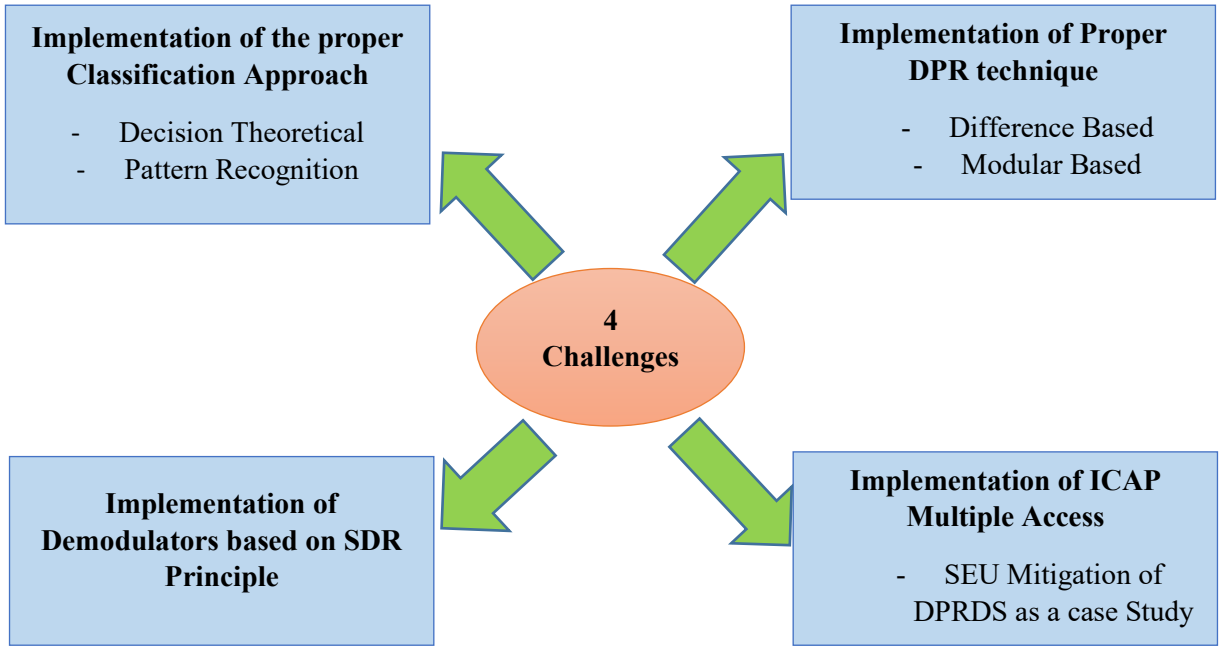


Figure 1-2. Dissertation Challenges

1.6. Dissertation Organization

The dissertation describes the methodology of developing DPRDS system based on FPGA DPR technique. The sequential of the dissertation explains the methods principles, techniques, and steps that required implementing the DPRDS and shows the obtained results. The dissertation is divided into 6 chapters as simply stated in Fig. 1-3.

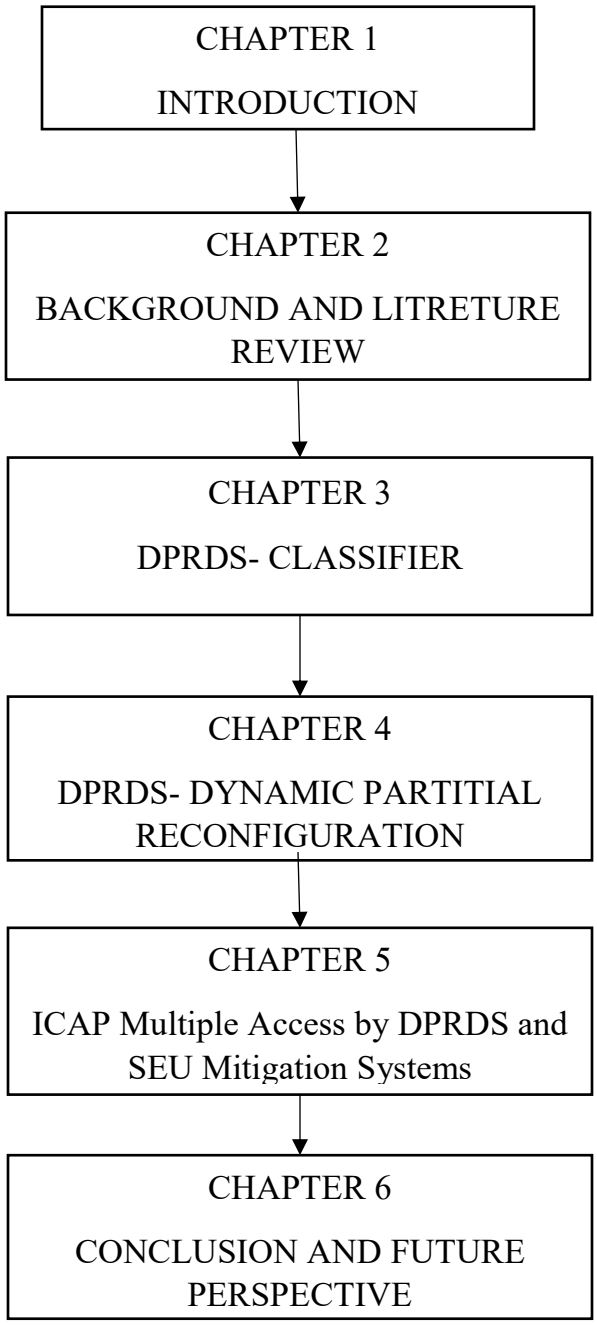


Figure 1-3. Dissertation Sequential

Chapter 1 is an introduction of the dissertation which describes the problem statement of the limited capabilities of GCSs/ satellites to communicate with several satellites/GCSs that use different modulation and demodulation techniques. This chapter describes also the motivation of this dissertation and the benefits of applying such research on real cases for both local and international zone. In addition, the objective, originality, methodology, and challenges of the research would be decribed. In methodology, the sequence of the implementation of the DPRDS and how get the benefits of merging the use of MATLAB as an auxiliary tool beside VIVADO tool to reduce the resources utilization and power consumption by implementing the complex system such as the training of the classifier using MATLAB and by employing the resulted parameters and coefficients in VIVADO tool.

Chapter 2 presents the background related to SDR principle and signal processing functions replaced or implemented by the means of software. It refers to the fundamental of FPGA architecture, definitions and methods that related to programing or configuring FPGA. It overviews the FPGA partial reconfiguration technique in terms of dynamically partial reconfiguration methods and the processes of generating the configuration bit-stream or the image that required for partial reconfiguration process. It presents the previous studies and researches related to this dissertation as much as possible, to show the new methods and techniques that are used to develop reconfigurable demodulators.

Chapter 3 describes the classification part of the DPRDS system. It presents the evaluation system that generates BPSK and QPSK modulations to test the performance of the DPRDS system. In addition to the classification approach, this chapter explains how to extract features from the signal and to implement a SVM classifier including kernel function and learning method using both MATLAB and VIVADO tools, in order to implement an efficient system with high recognition rate, low power consumption and less resources utilization. It presents the results of the classification part in the presence of AWGN and comparison with the results of the previous studies as well as the consumed classification time.

Chapter 4 describes the DPR part of the DPRDS system. It presents the partial reconfiguration approaches, dynamic partial reconfiguration implementing methods, and the procedures required implementing full and partial configuration bit-stream files that are used to configure FPGA before and during runtime. As well as, the communication method between partial reconfiguration controller and the storage memory would be explained. The results show the reconfiguration time required to reconfigure the dynamic section of FPGA with the corresponding reconfigurable module as BPSK or QPSK demodulator.

Chapter 5 describes the ability of sharing ICAP primitive among several user applications. It presents the ICAP multiple access by DPRDS and Soft Error Mitigation (SEM) application. It shows the method to inject and mitigate errors into static and dynamic parts of DPRDS design without any interruption to the reconfiguring performance. It presents a new error injection method that can reduce the error injection time by about 96% comparing to the traditional method. The results show that both systems are working properly without any effects on their performance.

Chapter 6 concludes the purposes and results of the dissertation and discusses the feasibility of developing such DPRDS systems in a generic receiver which promotes international cooperation. It presents the future perspective of the research by applying the generic receiver into a real GCSs and satellites.

Chapter 2 : Background and Literature Review

2.1. SDR Overview

2.1.1. SDR Background and Principle

Since 1890 when the first wireless transmission occurred, the wireless communication has been witnessing remarkable rapid development especially in transmission techniques. Passing through analog voice communication was transmitted by using limited bandwidths in 1930 [10], [11]. Then after 20 years the analog television broadcast required wider bandwidths was developed. Then, in the 60s, the computers are used as long-distance communication medium via the network system called Internet [12], [13]. Regarding to the continuous progress in communication field, Federal Communications Commission (FCC) has originated Wireless Local Area Network (WLAN) and organized the regulations of licensed and unlicensed spectrums, and allocated three regions for medicine, industry, and science [14]. Then the progress in wireless communication has been continued to investigate protocols as Wi-Fi and Bluetooth which nowadays were used in most network-based applications.

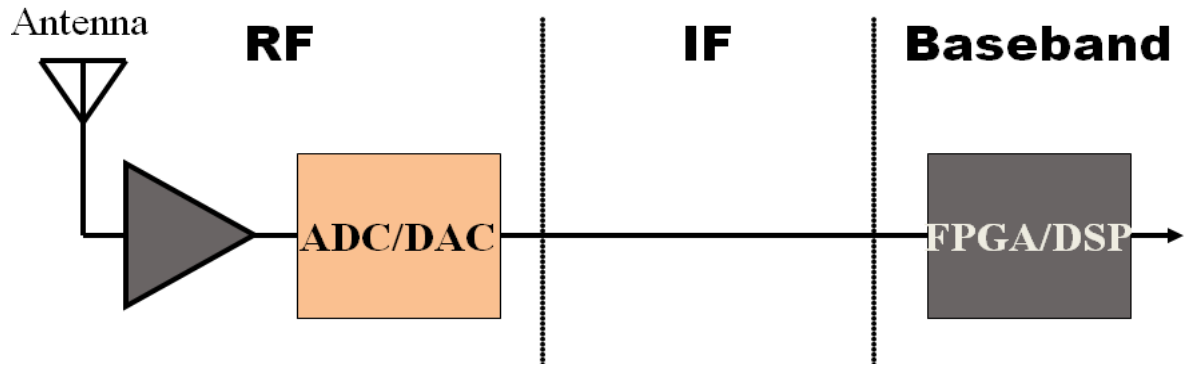
The rapid development and growth in wireless communications is evolving several technologies, but the main remarkable problem is most of the protocols and radios based on hardware, which reduce the flexibility of mitigating errors occurring in radios hardware. If there is an error happened in firmware or hardware, we cannot get suitable way to mitigate the errors. In that time, there is no visibility to reconfigure the devices by other protocols or software because the functionality of these devices is based on hardware component. These problems lead to think about a new concept and principle that can support and overcome the described problems and at the same time provide more benefits which is called now as SDR.

SDR is considered the base of generic wireless devices, where modulation schemes and some parameters such as carrier frequency and symbol rate, should be reconfigured or replaced via adaptive and reconfigurable communication system without any change in the platform. SDR can be defined as a radio communication where some or all hardware functions are implemented by means of software or software defined instead of hardware. It is a radio whose components are implemented in software rather than in hardware.

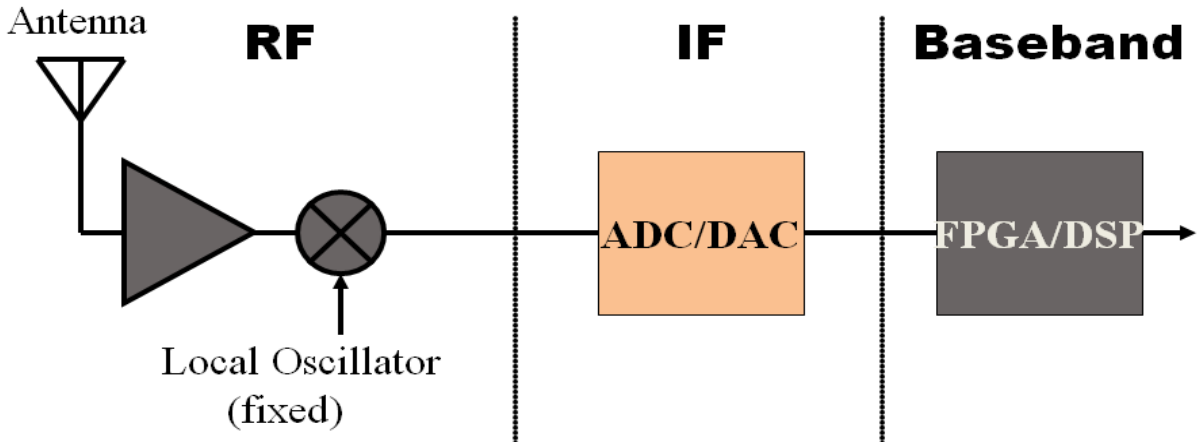
The principle of SDR is that more of the signal processing manipulation including filtering, modulation and demodulation functions are implemented in reprogrammable and reconfigurable digital hardware so that the platform can be used for several signal processing and SDR system provides portability and flexibility. Digital hardware that can be used for implementations of SDR modems include Field Programmable Gate Array (FPGA), Digital Signal Processor (DSP), and General Purpose Processor (GPP).

2.1.2. SDR Architecture

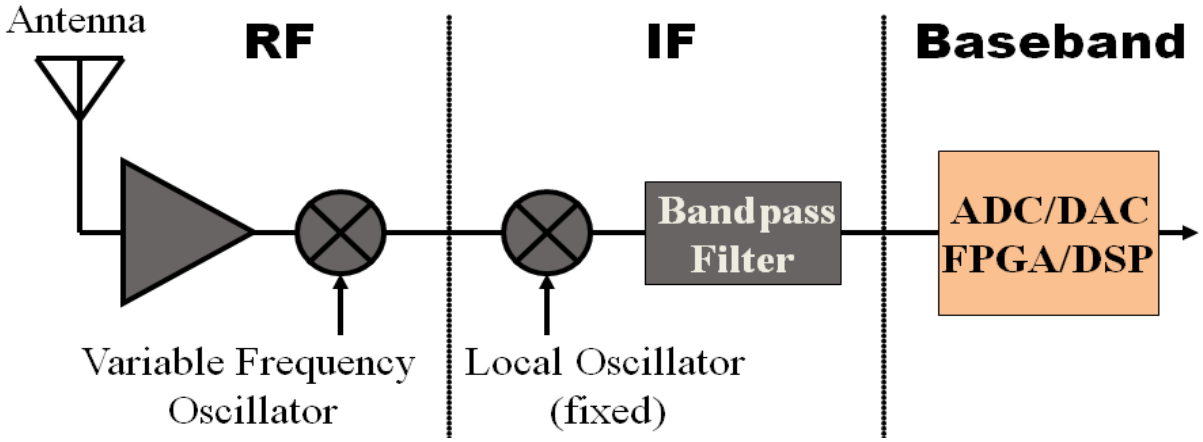
In SDRs most of the functionality is software defined without the need to equivalent hardware and most of the signal processing is digitally performed. Real SDR system is segmented into 3 stages; Radio Frequency (RF) front end, Intermediate Frequency (IF), and baseband signal processing stages. The radio analog signal is first received by the antenna, then the down conversion module converts the high frequency signal to IF signal, after that the IF signal is converted to digital signal through Analog to Digital Converter (ADC) to be digitally processing in baseband stage using FPGA, DSP or GPP. There are three SDR configurations depending on the down conversion and ADC stages, and the positions as can be seen in Fig. 2-1.



(a) ADC/DAC Implemented in RF Front End



(b) ADC/DAC Implemented in IF Stage



(c) ADC/DAC Implemented in Baseband Stage

Figure 2-1. Different Configurations of SDR System

A real SDR model can be described in Fig. 2-2, where the received signal is down converted to IF, and then the IF signal is filtered to prevent the existing of the aliasing frequency into the band of frequencies. The IF signal will be digitized by generating numerical values from the filter through ADC. After that the digitized data can be processed using FPGA, DSP, or GPP [15].

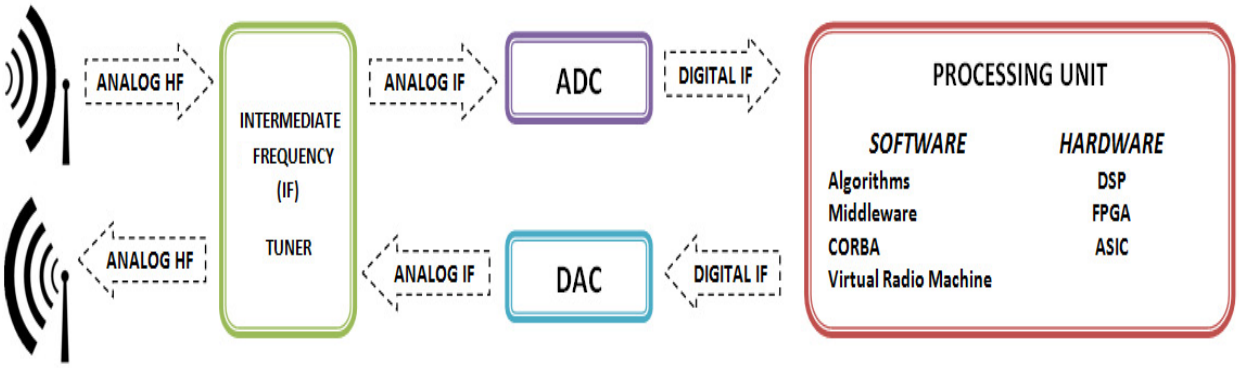


Figure 2-2. Real SDR Model [15]

For the transmission path, after applying the processing on digital data, the processed data is converted to analog signal after passing through Digital to Analog Converter (DAC), and then analog IF signal is up converted to high frequency and transmitted by antenna.

2.1.2.1. SDR Transmitter

As described previously the digital data is applied to the transmitter input side of an SDR system, and this data is generated from FPGA or any other signal processing devices. Digital Upconverter (DUC) translates the baseband signal to the IF signal. IF signal samples are converted into analog IF signal via DAC. Then the analog IF signal is converted to RF signal of high frequency by passing through RF upconverter. To transmit the generated signal to travel for long distance, power amplifier is used to increase signal energy as seen in Fig. 2-3 [16].

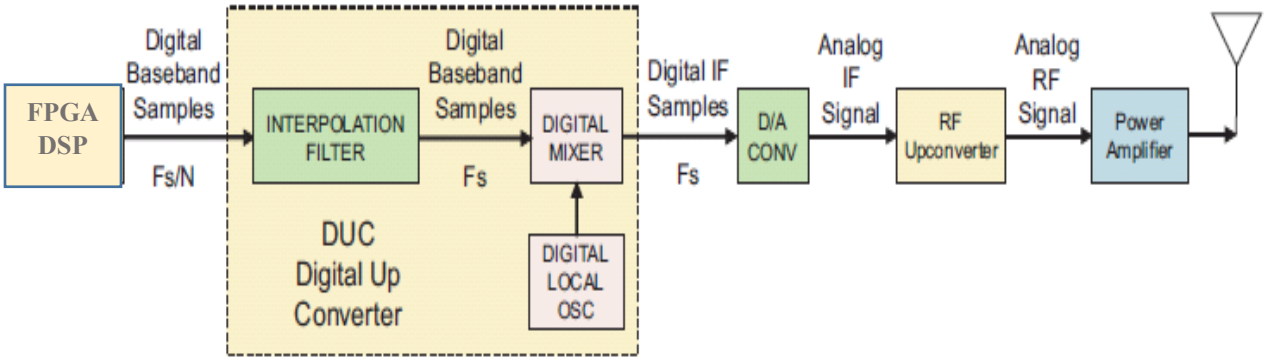


Figure 2-3. SDR Transmitter Block Diagram

2.1.2.2. SDR Receiver

The receiver part of the SDR system starts by converting received RF signal into IF signal before passing through ADC that converts analog IF signal into digital samples. These samples have to apply to Digital Downconverter (DDC) to be easily manipulated by the FPGA/DSP devices. The DDC is considered as the key of SDR system and is implemented by FPGA IP core or externally single monolithic chip. The Finite Impulse Response (FIR) filter is used as low pass filter with decimation and limits the bandwidth of the dwonconverted signal. Therefore FPGA/DSP can perform the required signal processing such as demodulation, decoding, and other processing as seen in Fig. 2-4 [16].

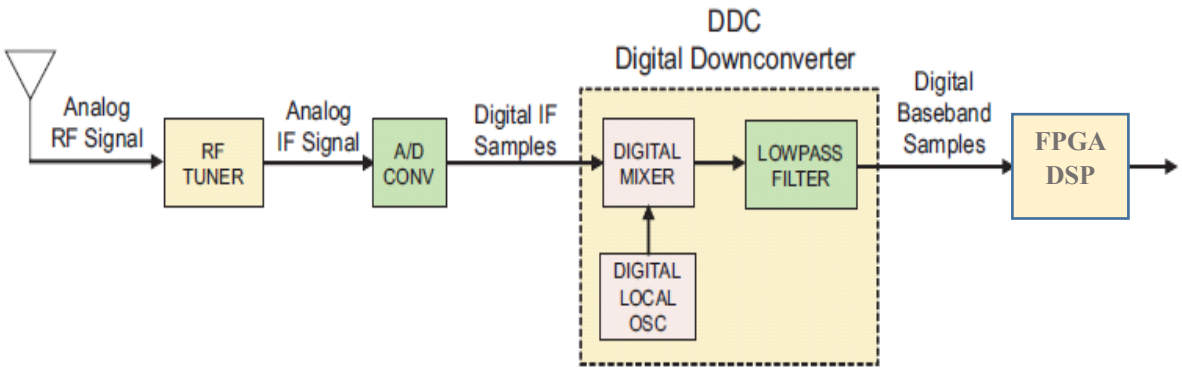


Figure 2-4. SDR Receiver Block Diagram

2.1.3. SDR Platforms

FPGA, DSP, and Application Specific Integrated Circuits (ASIC) are considered as hardware alternatives that can be used to implement SDR system. A comparison among these chips has to perform for taking the suitable chip selection to the corresponding application. The comparison aspects should cover power consumption, Resources utilization, cost, and other aspects that can help for making good selection as shown in Table 2-1 and Fig. 2-5.

Table 2-1. Comparison among Different SDR Platform Chips

Comp. aspects\Type	FPGAs	High Speed DSPs	GPPs	Multiple ASICs
Power Consumption	Low	Very High	Moderate	Very Low
Resources Utilization	Low	Moderate	Moderate	Large
Cost	Moderate/Low	Moderate/High	Moderate	High
Field Upgradable	High	High	Some	None
Silicon Evolution	Easy	Easy	Moderate	Available

From Table 2-1, using FPGA is the most suitable selection to implement SDR system especially, and it contains DSPs blocks that can be reconfigured and used for parallel computations. DSP is coming in the second ranks after FPGA because it can perform several signal processing applications and has microprocessors with architecture, instructions and features [17].

Several SDR platforms have been developed to support different research projects. The most five popular SDR platforms are Universal Software Radio Peripheral (USRP), GNU Radio, Kansas University Agile Radio (KUAR), The Japanese National Institute of Information and Communications Technology (NICT) SDR Platform, and Berkeley Cognitive Radio Platform.

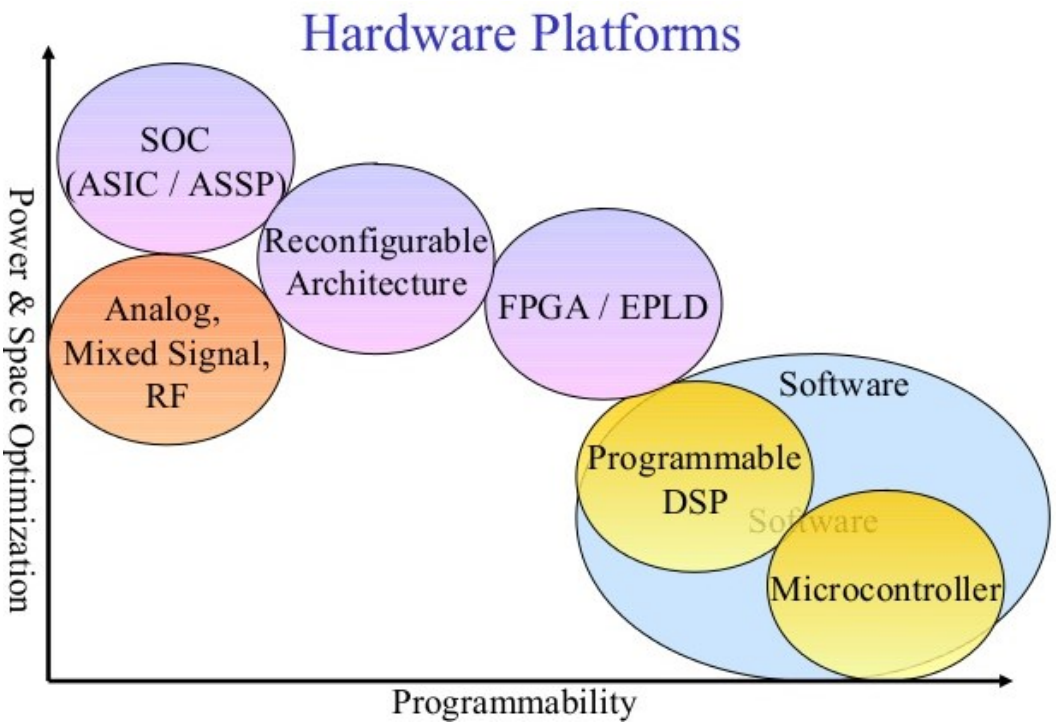


Figure 2-5. Comparison among FPGA and other programmable devices

The USRP is SDR platform that is currently available and provides the GNU Radio with hardware platform. USRP can support up to 25 MHz of bandwidth and is considered mostly for experimentation because it does not match any communication standards. KUAR is the developed form of the GNU Radio to be a low-cost experimental SDR platform and can operate with frequencies range of 5.25 to 5.85 GHz with bandwidth margin of 25 MHz. NICT SDR platform is developed to test mobile networks and supports frequencies range from 1.9 to 2.4 and 5.0 to 5.3 GHz. And it manges the selection among existing standards. These SDR platforms are based mainly on FPGA and combination of CPU [18].

The advantage of using SDR platform especially for space application sector is to provide the reconfigurability and flexibility to communication subsystem as reduced cost, and Commercial Off-the Shelf (COTS) components due to fast development times. The combination of both SDR communication platform and SOC increases the reduction in both cost and mass.

2.2. Automatic Digital Modulation Recognition Overview

2.2.1. ADMR Principle and Background

Automatic modulation recognition (ADMR) technique objective is to recognize the modulation scheme of a received signal with a high probability of recognition rate within a minimum observation time. Two methods can be used to realize ADMR. One is based on a prior knowledge of the received signal parameters and the other which is called blind recognition method which does not need any prior knowledge of the received signal parameters to identify the modulation schemes. The blind recognition method identify the type of modulation from the information existing in the few possible number of received samples, and according to statistical inference based on errors in received signal, the modulation type can be identified. But the blind recognition method is more complex than the other method, therefore in this dissertation the unblind recognition method is used.

ADMR is considered to become one of the important applications in SDR receivers that gives the ability to change the demodulator of the receiver system according to the recognition of the received modulated. It has to identify the modulation scheme of received signal. So it plays an important role between detection and demodulation stages. ADMR is an intermediate step between signal identification and demodulation processes, and plays an important role in several civilian and military applications. ADMR is the key role in the implementation of advanced wireless communication system especially for satellite communication system. Since 1980s up to date, the interest in modulation recognition has been growing. It has several applications such as spectrum management, interference identification, monitoring, surveillance, and signal confirmation [19], [20]. Recently, SDR and reconfigurable communication system are the most attractive application scope.

ADMR is extremely important in communication intelligent applications for several reasons. Firstly, applying the signal to an improper demodulator may partially or

completely damage the signal information content. Secondly, knowing the correct modulation type helps to recognize the threat and to determine the suitable jamming waveform.

Modulation recognition is an important part in SDR, where the modulation scheme can be varied according to channel capacity, and the modulation scheme can be detected at receiver side in real time using automatic modulation recognition [21]. For all communication systems, signals should be safely transmitted and received, whereas noise and unwanted signals must be defined, identified, and jammed [22], because of the complexity of blind recognition method, Usually, auxiliary information is combined into transmitted signal to reconfigure the SDR system.

Block diagram of the communication system model can be simply drawn as shown in Fig. 2-6 [23]. The modulator converts input symbols to signal waveform, and then the channel model and jamming is applied on modulated signal for the transmission. At receiver side the noise is added to the received signal before the classification stage. The classification stage contains of two steps; Signal preprocessing and selection of the proper classification algorithm. Preprocessing includes some or all of, noise reduction, estimation of carrier phase and frequency, signal power, and symbol period, etc.

According to the selected the classification algorithm, the accuracy of preprocessing tasks varies; some classification algorithms require high precision, whereas others are less sensitive to the unknown parameters. Then after the proper classification of the modulation scheme of the received signal, the received signal is applied on the corresponding demodulator. Fig. 2-6 illustrates the block diagram and the sequential processes applied on the data or symbols that required to be transmitted till the regeneration of these symbols in the receiver side [23].

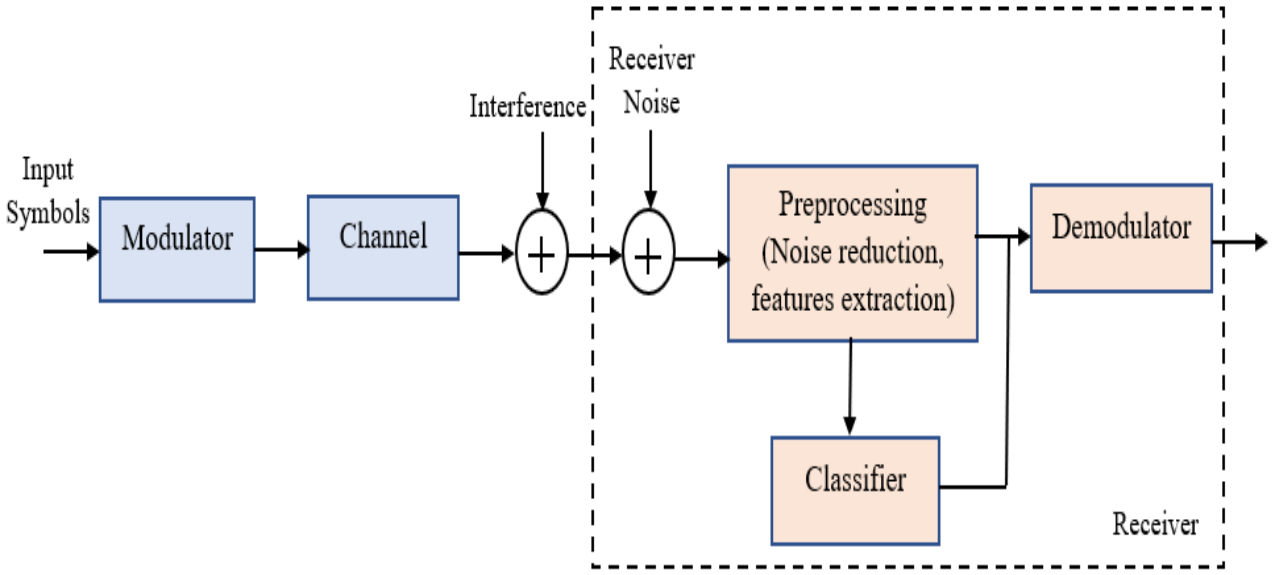


Figure 2-6. Basic pattern recognition system Block Diagram.

Two popular approaches are used to implement the classifier of received modulated signals; the Decision-Theoretic (DT) approach based on likelihood function, and the Pattern Recognition (PR) approach based on extracting unique measurable values (features) of the signal. DT approach based on multiple hypothesis testing has high performance in terms of correct classification percentage (optimal), but the drawbacks of this approach are high computational complexity which leads to that the classifier is impractical, and the sensitivity to impairments as frequency and phase offset.

On the other hand, PR approach based on pattern matching is suboptimal, simple to implement, and robust. PR consists of two stages. The feature extraction and classifier stages, Feature extraction stage extracts distinctive and small amount of information from the received signal, and classifier uses the distinctive information to classify the modulation scheme of the received signal. In this dissertation, pattern recognition approach is used [24].

2.2.2. ADMR Approaches

2.2.2.1 Decision-Theoretic

In this approach, all classification methods is mainly based on likelihood function which is achieved mainly by statistics calculation. It is a function of transmitted data and channel parameters. The likelihood function is computed for each hypothesis, modulation schemes in this dissertation, by using the equally likely priors assumption, and the modulation scheme that maximizes the likelihood function is the decision result of the classification process. The features based method requires designers to select distinctive features of the signal that could represent each modulation scheme in a unique manner. Based on the observation of the features values the decision is taken.

The likelihood function based algorithm minimizes the probability of error of classification which makes this method optimal in the Bayesian sense. However, the defect of this method is that sometimes the evaluation of likelihood function could become complex and very messy because of the absence of knowledge of parameters, or channel conditions, which makes the likelihood based method impractical. Compared to the optimal one, a suboptimal method becomes a good choice with reasonable computation complexity [25].

2.2.2.2 Pattern Recognition

In this approach, the main key points are the features and the classifier. The modulation classification based on this approach is divided into three functional blocks, namely the pre-processing, feature extraction, and the classifier blocks as shown in Fig. 2-7 [26].

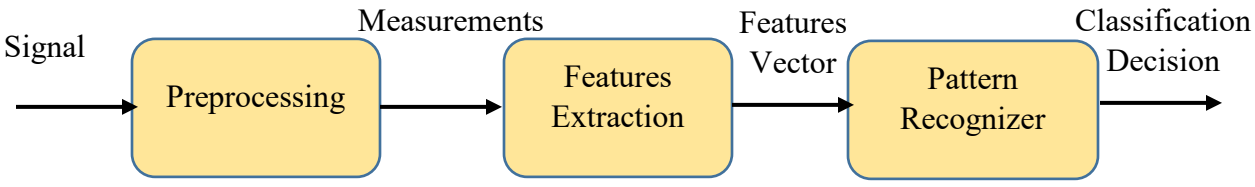


Figure 2-7. Pattern Recognition System Block Diagram.

2.2.2.2.1. Pre-Processing

A noise reduction, filtering, encoding and other preprocessing functions can be applied on the signal in this stage for the enhancement of extracting the features vectors.

2.2.2.2.2 Features Extraction

Features are any extractable measurements that can represent the input signal. The feature extraction part is used to reduce the dimension of the measurement by extracting the distinctive features which should be simple and fast to be calculated. Applying feature extraction to the input data can result in features as well [26].

Feature extraction keys are extracted from the radio signal. Some of the commonly feature extraction keys are a constellation shape recovery method, Discrete Wavelet Transform (DWT) and Discrete Sine and Cosine Transform (DST, DCT), higher-order statistics (HOS), including moments, cumulants, and cyclic cumulants (CC) of the signal, and usage of information contained in the received signal.

2.2.2.2.3 Pattern Recognizer

Pattern recognizer processes those extracted features to identify the modulation scheme of the received signal according to a pre-designed decision rule. The popular pattern recognizer methods used for classification purposes are SVM, Multi-Layer Perceptron Neural Network (MLPNN), and fuzzy logic [26].

Table 2-2 shows a comparison between DT and PR approaches; the PR approach is the selected approach to be used in the design and implementation of the automatic modulation classifiers section for this research work according to the advantages mentioned in Table 2-2. In this research, the features extracted from discrete transform as DWT are used as feature extraction keys, and DWT not only extracts features from the input signal, but also reduces features dimensions by extracting these features after three stages of DWT. SVM is used for the development of the classifier of the classification section of the DPRDS. Details on the development of the AMR or classification section and DPR of the DPRDS system for this study work will be presented in the next chapters.

Table 2-2. Comparison between AMR Approaches

Comparison Aspects	Decision -Theoretic	Pattern Recognition
Principle	Based on the likelihood function.	Based on Features.
Methods	- Statistical moments. - Likelihood functions. - autoregressive spectrum Modeling.	- Pre-processing of signals. - Key feature extraction. - Classifier.
Classification	Based on multiple hypothesis tests (with threshold values).	Based on Pattern Matching
Advantages	- Optimal (minimizes the probability of false classification) - Sub-optimal (Hard to implement) - Better performance in terms of correct classification percentage.	- Nearly optimal. - Easy to implement. - Classifying more modulation schemes than DT. - Robust with respect to model mismatch. - Don't need such hypothesis testing. - a threshold is chosen automatically and adaptively unlike the other approach
Disadvantages	- High computational complexity (classifier Impractical). - Difficult to implement. - Not a viable option in most real-time scenarios. - Difficulties in forming the right hypothesis testing. - Not robust with respect to model mis-match	Non-optimal.

2.3. FPGA Overview

The similarity of the architecture of FPGA and ASIC devices put them in a competition with the increment of implementing fast and complex computational designs especially for satellite communication systems. ASIC device is designed to realize a specific application, which can be considered as an advantage and disadvantage, where the disadvantage is to be dedicated for particular application and cannot be used to realize other applications which reduces its flexibility, and the advantage is to utilize resources and power consumption is low. On the other hand, FPGA device is not dedicated for a specific application and can be used to realize several applications which increase the flexibility and programmability. The resources utilization and power consumption of FPGA is based on the implemented application and differs from one application to another.

For the development of generic receiver for digital modulation techniques, it is required to implement a cost-effective FPGA-based design with less resource utilization and low power consumption. Xilinx provides DPR feature of FPGA since 2003 [27] which increases the flexibility and optimization of the design, so as to reduce both of used resources and consume power. In this research, applying DPR technique of FPGA to realize and develop a DPRDS prototype of a generic receiver is described. Development methodology, design steps, and DPR realization of the DPRDS prototype are proposed and tested.

Design architecture and methodology using DPR of FPGA that used to implement DPRDS prototype will be introduced in the following chapters. Chapter 3 and 4 provide explanations to understand architectures, layers, configuration methods, and DPR methodology of FPGA.

2.3.1. FPGA Architecture

The architecture of FPGAs varies according to FPGA family and manufacturing company, but it can be stated that the common FPGA architecture consists of three main sections; Configurable Logic Blocks (CLB), Configurable I/O Blocks, and programmable interconnect as seen in Fig. 2-8. In addition, there are a Mixed Mode Clock Manager (MMCM) and Phase Locked Loop (PLL) that contain clocking circuit to achieve clocking requirements of the user design. Several memory cell types can be used into FPGA, Static Random Access Memory (S-RAM), anti-fuse, and flash memories. Configurability-wise, SRAM is the most popular FPGA memory cell type that is used for several applications.

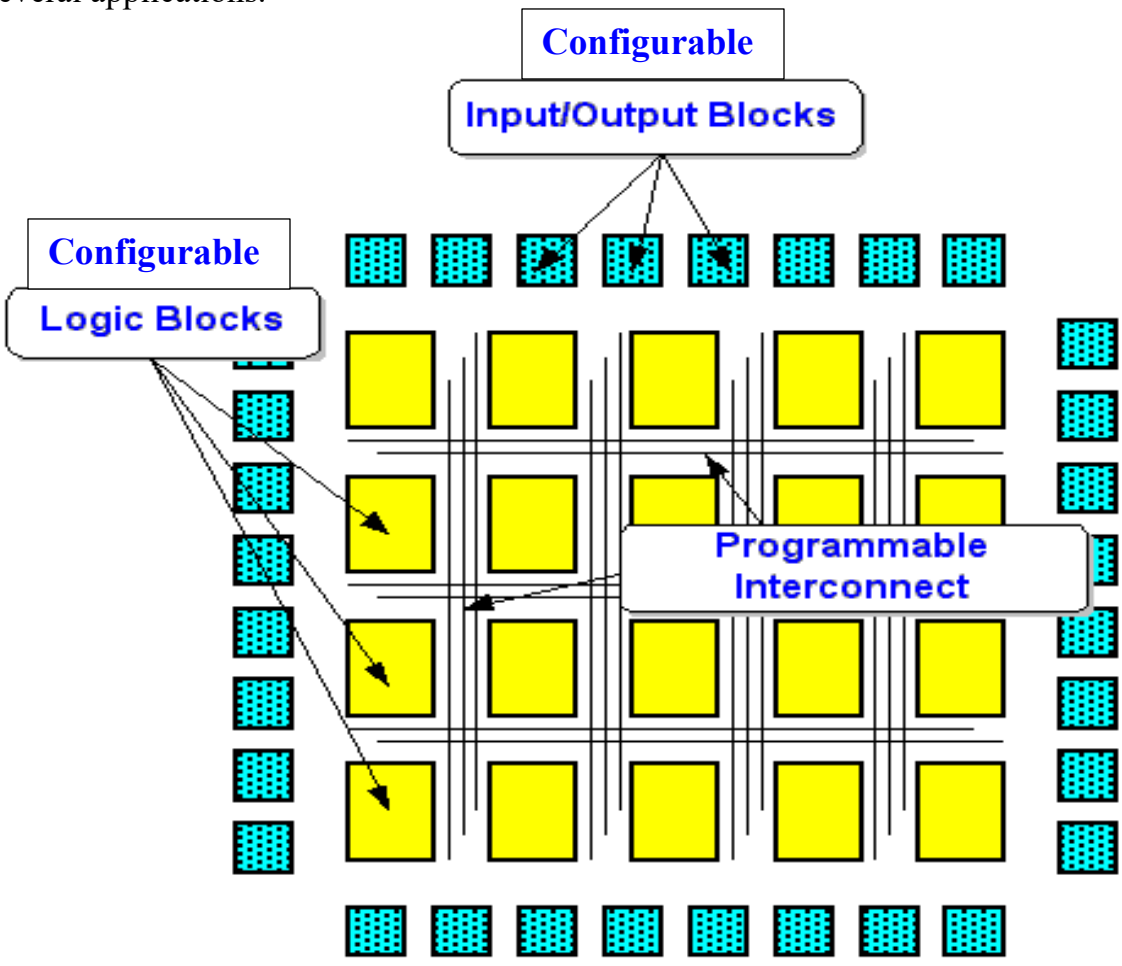


Figure 2-8. Generic FPGA Architecture.

2.3.1.1. Configurable Logic Blocks

CLBs contains the programmable logics that are configured and programmed according to user design. Each CLB contains a number of slices and each slice contains Lookup Tables (LUTs), Flip Flop (FFs), and multiplexers. The communication among several CLBs is performed via switching block.

CLBs of Xilinx 7-family FPGA provide high performance logics and support sequential and combinational functions which contain 6–input LUT, shift register and distributed memory, high speed Arithmetic Logic Units (ALU), Dual 5-input LUT, and wide multiplexers as seen in Fig. 2-9, and 2-10 [28].

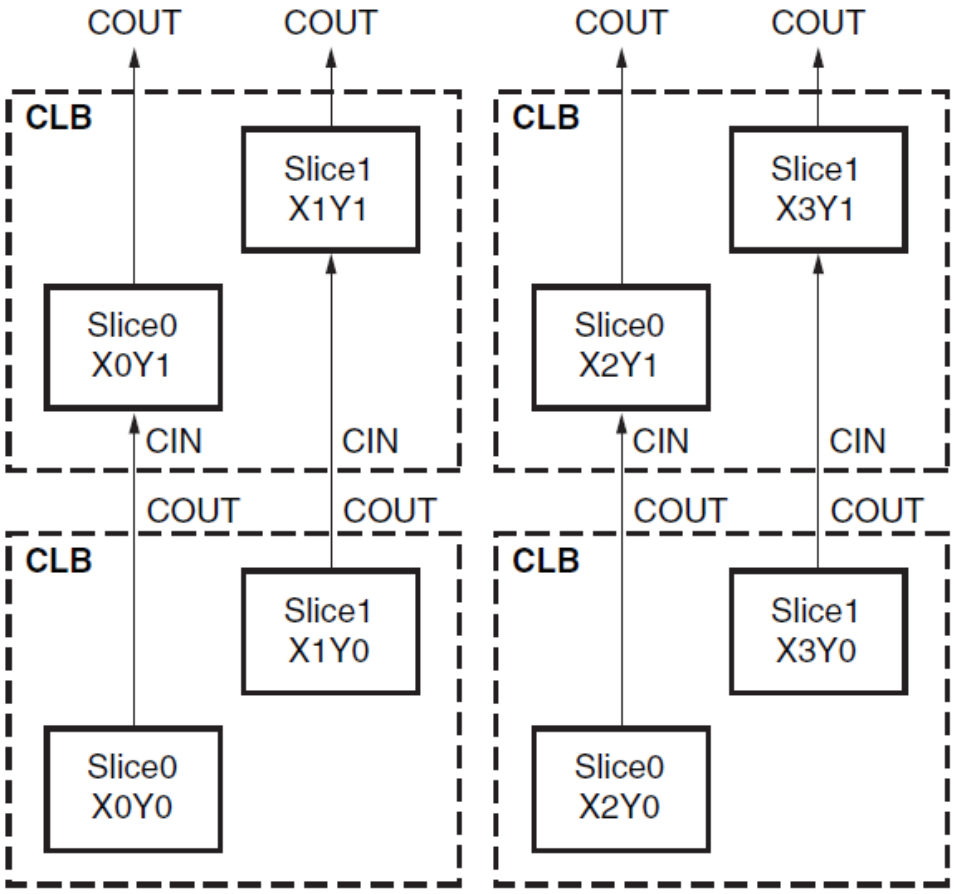


Figure 2-9. CLB Architecture [28].

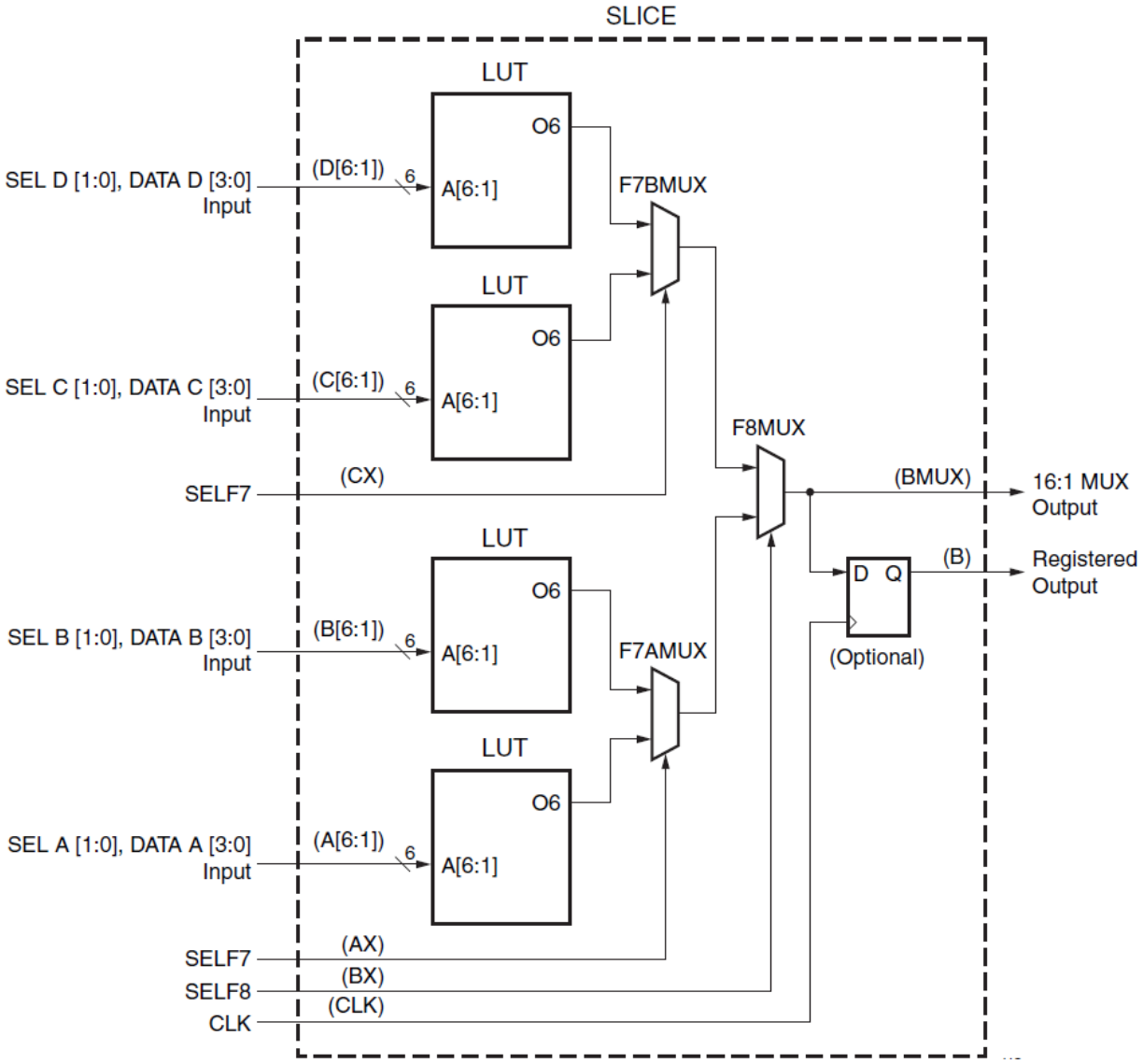


Figure 2-10. Slice Block Diagram [28].

2.3.1.2. I/O Blocks

I/O blocks are considered as the intermediate interface between the inner user design input/output parameters with the outside environment. They have two I/O types; high performance and high range types. High performance type has I/O delay capability and supports I/O standard up to 1.8 V, and high range type supports I/O standard up to 3.3 V. Fig.2-11 illustrates the architecture of FPGA I/O block.

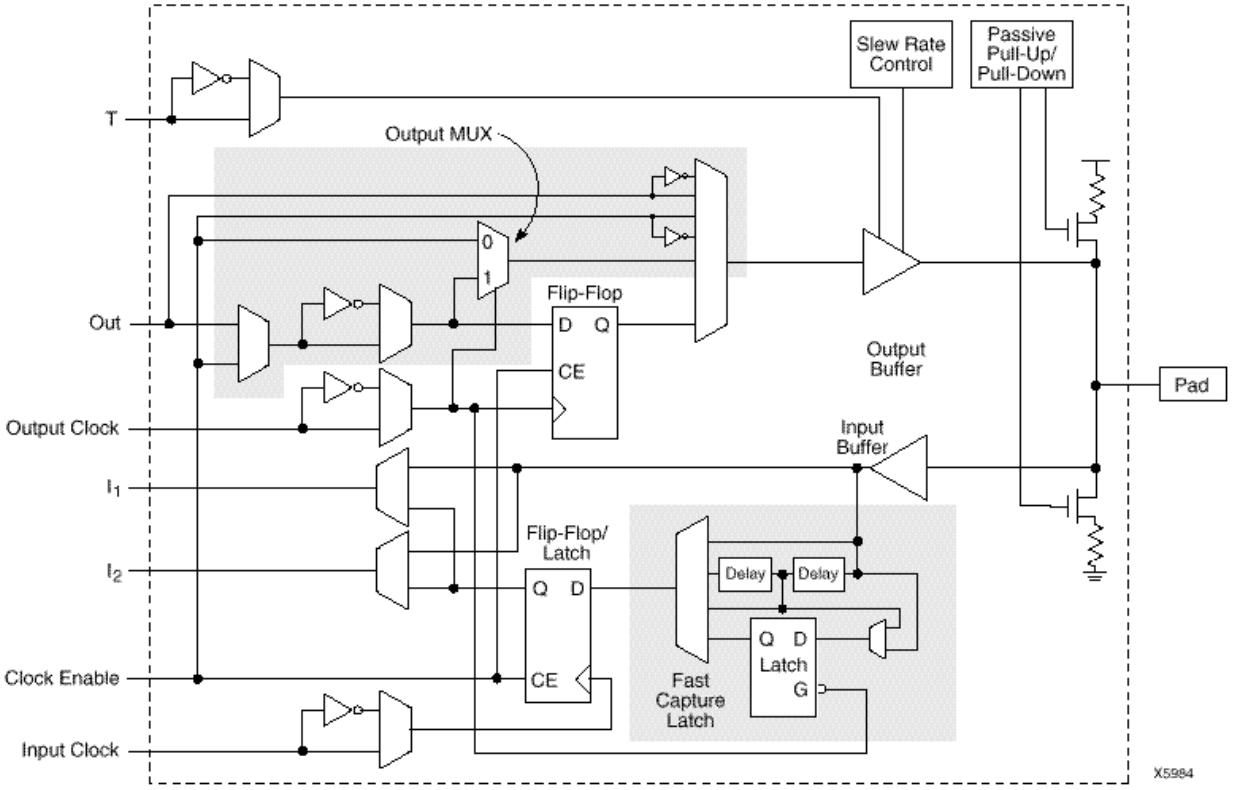


Figure 2-11. I/O Block Architecture of XC4000EX Series.

2.3.1.3. Programmable Interconnect

The FPGA programmable interconnect is used to connect CLB blocks together to exchange data among them. There are two types of programmable interconnect; long and short lines. Long line is used to connect CLB blocks that are far from each other with less possible delay, and short line is used to connect closest CLB blocks together. Switching matrices connect both long and short lines together. Programmable switches act as intermediate connection hub where it attaches CLBs to each other through interconnect lines, and the interconnect lines together and to the switch matrices as seen in Fig. 2-12.

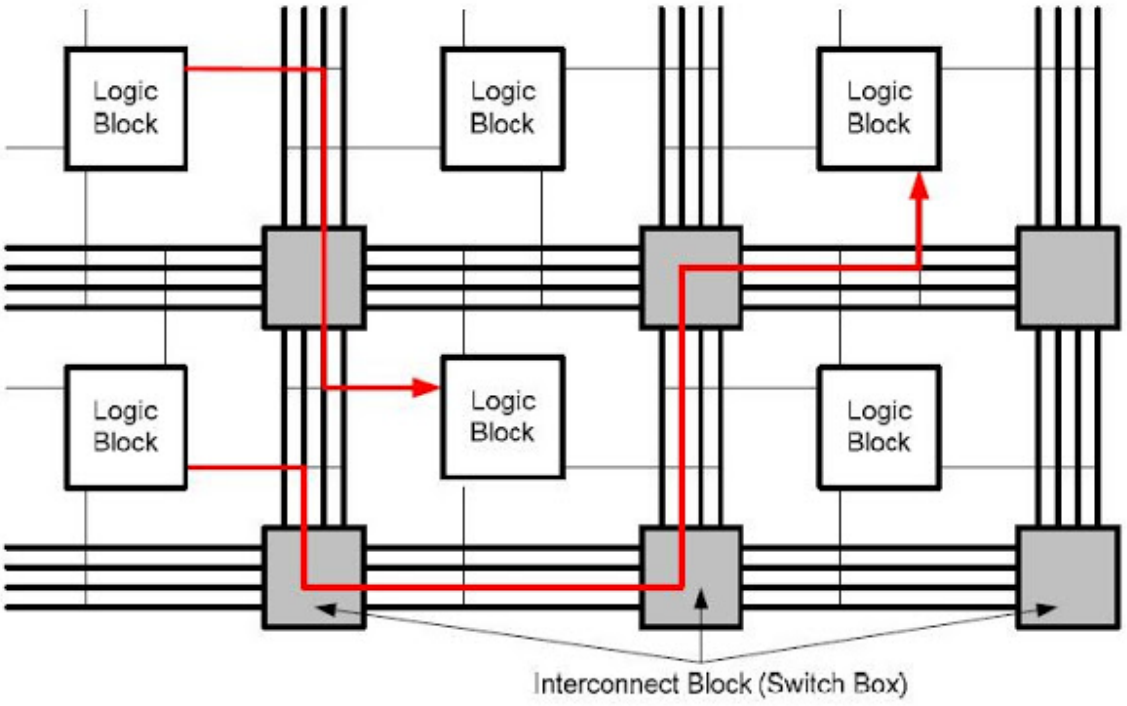


Figure 2-12. Programmable interconnect with Switch Matrix.

In all Xilinx 7 series FPGAs, good distribution and alignment of the common elements inside FPGA lead to enable easy use and reuse of these elements and IPs for fast design probability and scalability form low cost to high performance. For example, common elements alignment of Kintex-7 FPGA can be seen in Fig. 2-13.

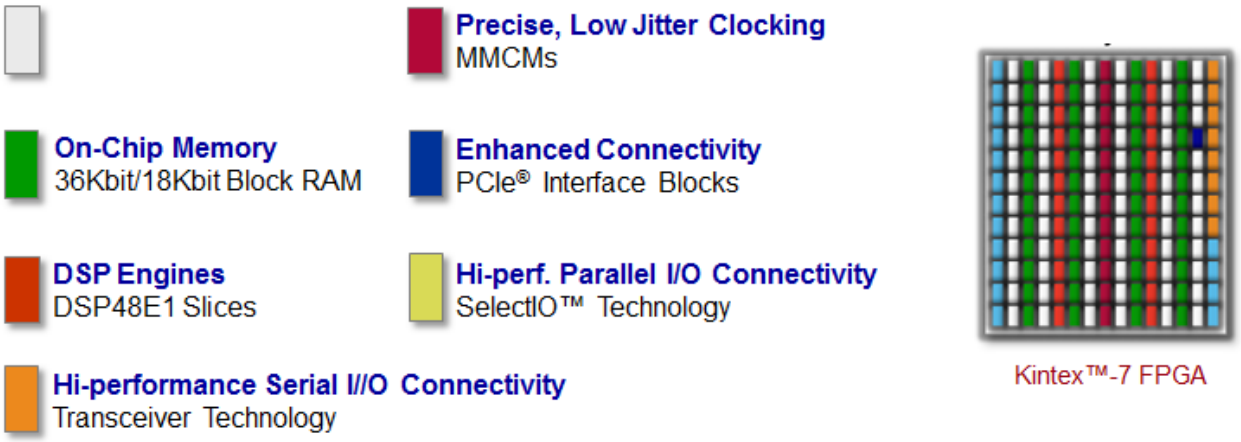


Figure 2-13. FPGA Architecture Alignment.

2.3.2. Application Layer

From the FPGA architecture description, FPGA contains several resources to allow the users to realize their applications. It contains multipliers declared as Digital Signal Processor (DSP) and DSP48 which performs signal processing operations, LUTs, FFs, Block Random Access Memory (BRAM), and other components. All these resources can be configured and working together to realize user application.

Programmable interconnect exists in the application layer and its role is to interconnect FPGA resources to each other through switching boxes, and to the outside environment through I/O blocks and high speed Multi-Gigabit serial I/O transceiver. Fig. 2-14 illustrates the contents of the application layer.

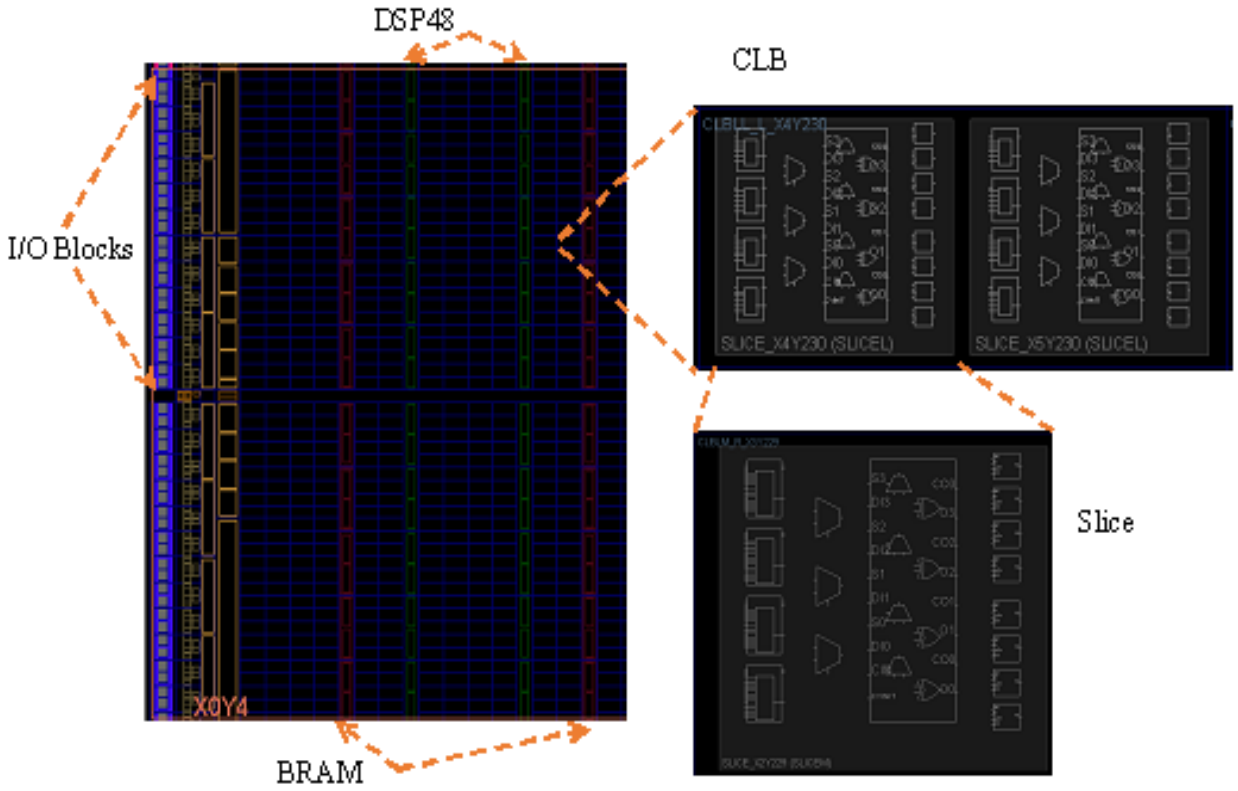


Figure 2-14. FPGA Application Layer Elements.

2.3.3. Configuration Layer

Configuration layer controls the configuration of the FPGA application layer. The configuration and memory state of the application layer are controlled by the configuration layer of the FPGA. After the configuration process is performed, the FPGA application layer will match the user design circuitry. FPGA contains several configuration ports, it has a JTAG configuration access port, a selectable microprocessor access port (SelectMAP), and an International Configuration Access Port (ICAP). All these ports can be accessed by Configuration Packet Processor (CPP) with the written data on these ports. After receiving the written data from access ports, CPP performs read and write operations to the Frame Data Register (FDR) and control registers. Writing data to control register as the self-reconfiguration process of FPGA can be performed according to a trigger from the designed code. In addition, FPGA provides also peripheral interfaces; a Serial Peripheral Interface (SPI) and a Byte Peripheral Interface (BPI).

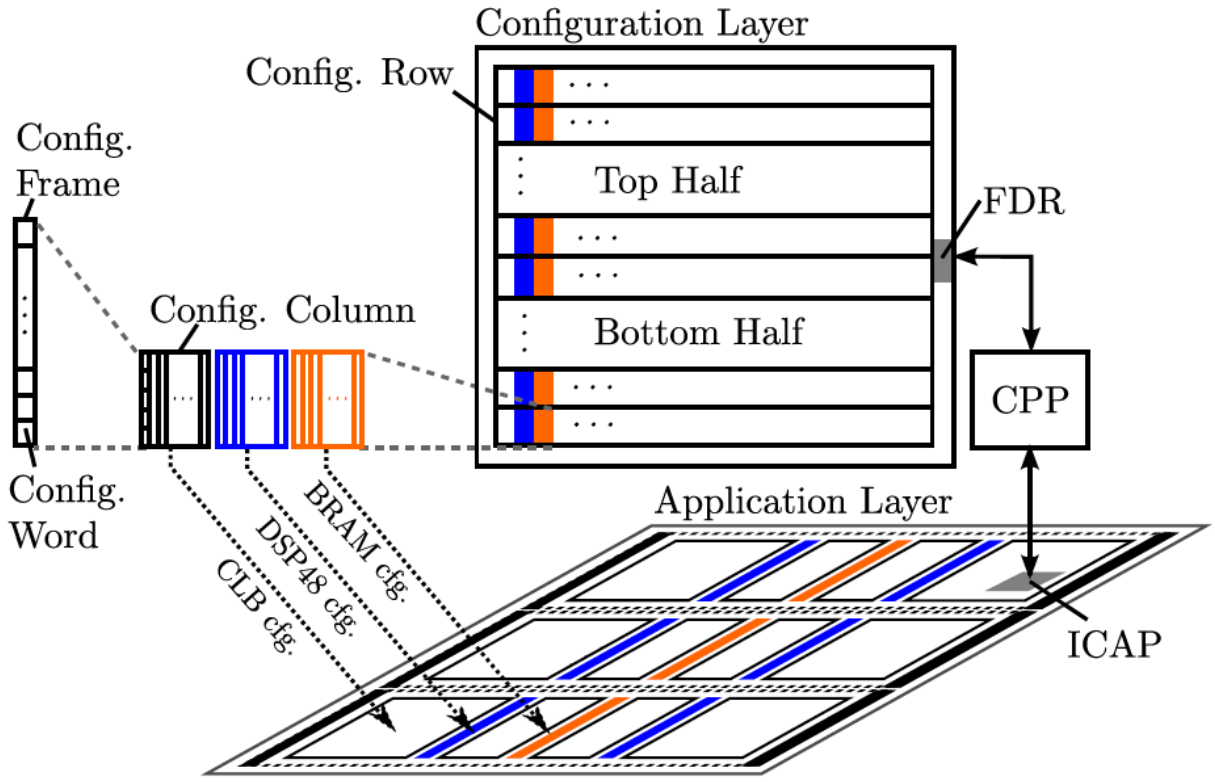


Figure 2-15. Concept of Operation of Configuration and Application Layers [27].

The configuration layer is segmented into several configuration rows. These rows can be divided into top-half and bottom-half rows as seen in Fig. 2-15 [27]. Each configuration row has different configuration columns and each column is related to different resource elements, such as DSP48 slices, BRAMs, CLBs, and IOBs. Each configuration column contains a number of configuration frames, and each configuration frame consists of configuration words. A configuration frame contains data that can affect several resources in the application layer. For example, a configuration bitstream required configuring FPGA is first transferred to CPP through ICAP. Then after the synchronization process, CPP will transfer the received configuration bitstream information to FDR until the transferring of the whole frame data. Data written in FDR is transferred to the FPGA configuration memory and in turns the application layer will be changed. For testing purposes, configuration frames can be read backed through the FDR output [27].

2.3.4. FPGA Configuration Methods

Several methods can be used to configure or program FPGA devices. There are full configuration, partial, dynamic partial, and self-reconfiguration methods. The selection of the configuration method is based on the application and objectives of user design circuitry.

2.3.4.1. Full Configuration

In full configuration, the application layer of FPGA is configured with a configuration bit-stream file (user-design) which means an existing design (application layer configuration) is replaced with another design and the configuration of the application layer. The current design will be changed according to the new user design. This type of configuration can be performed using JTAG, serial configuration, and selectMap ports.

2.3.4.2. Partial Configuration and Reconfiguration

Part or more of application layer of FPGA can be changed by modifying some of the configuration frames using partial configuration bitstream file. In partial reconfiguration designs, the designed code is segmented into two parts, one is the static part of the design

which stays fixed during runtime, and the other is the dynamic section which is the part of the design that can be changed during runtime without any effects on the performance of the static part of a user design. The dynamic part of the application layer is called reconfigurable partition. For some applications, the partial reconfiguration process can be performed by the user through JTAG port.

2.3.4.2.1. Dynamic Partial Reconfiguration

In dynamic partial reconfiguration, the configuration of a part of application layer can be changed during runtime without interrupting the performance of other parts, but this configuration type should be controlled by outside microprocessor and performed through ICAP port.

2.3.4.2.2. Dynamic Partial Self-Reconfiguration

In dynamic partial self-reconfiguration, the same concept of dynamic partial reconfiguration method but the difference of this configuration is to be performed internally FPGA, i.e. the reconfiguration process is controlled by an FPGA embedded microprocessor and the reconfiguration is performed through ICAP. In this dissertation, dynamic partial self-reconfiguration method is used. The simplified partial configuration process can be seen in Fig. 2-16.

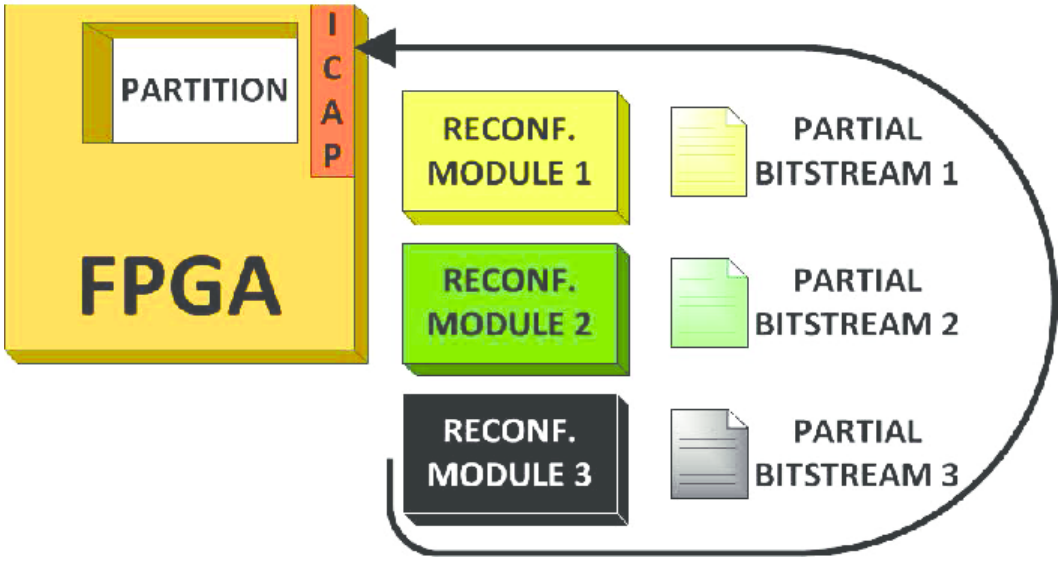


Figure 2-16. Simplified Partial Reconfiguration Concept.

2.3.4.1. Advantages of Partial Reconfiguration

The main advantage of FPGA partial reconfiguration that leads to the other advantage is the reduction of the amount of FPGA resources required to implement a user design, therefore , the design concludes reductions in both cost and power consumption. Reduction in the usage of FPGA resources provide flexibility in the selectivity among several algorithms, protocols, and techniques available to an application. This flexibility enables new types of FPGA designs that would be otherwise impossible to be realized. In addition, partial reconfiguration provides IP reuse and reduces the latency by spreading more area on submodules which implies high acceleration. Some of popular partial reconfiguration applications are FPGA fault tolerance, Single Event Upset (SEU) mitigation, and configurable computing acceleration.

2.4. Digital Modulation Techniques

Several digital modulation techniques are used in satellite telecommunication systems because of its advantages over analog modulation in terms of noise immunity, ease of multiplexing, security, and transmission error detection and correction. In digital modulation techniques, the digital signal (discrete in time and amplitude) modulates the carrier signal which is resulting in changing of some properties of the carrier signal as amplitude, frequency, and phase. The popular digital modulation techniques used for satellites telecommunication systems are M-ary Frequency Shift Keying (MFSK), Gaussian Minimum Shift Keying (GMSK), MSK, and M-ary Phase Shift Keying (MPSK).

In this dissertation, two types of MPSK modulations are used; Binary Phase Shift Keying (BPSK), and Quadrature Phase Shift Keying (QPSK). BPSK and QPSK are an M-ary constant amplitude digital modulation schemes where M equal to the number of different phases, symbols or output, that will be transmitted according to the change of one or two successive bits in the bit stream (input data), for BPSK M=2, and for QPSK M=4 respectively. An overview on both BPSK and QPSK modulation techniques is introduced in the next few sections in this chapter.

2.4.1. BPSK Concept

In BPSK modulation, the input binary data required to be transmitted through transmission medium are represented by two sinewave symbols of phases 0° and 180°. This section describes both modulation and demodulation processes overview of BPSK modulation type.

2.4.1.1. BPSK Modulation

Every input binary data (1 or 0) has to be passing through Non Return to Zero (NRZ) block to convert these binary data to 1 or -1data respectively. After that, these data have to be multiplied by a carrier signal to generate a BPSK modulated signal [29]. For each input binary data, one symbol will be generated; one of the generated symbols of 0° phase shift and the other of 180° phase shift. The mathematical expression of BPSK generated signal is in Eq. (2-1).

Fig. 2-17, shows the block diagram required to generate BPSK modulated signal to be ready for the transmission through communication medium. Fig. 2-18 illustrates the generation processes of BPSK modulation, i.e., the corresponding output symbol according to each input binary data. Fig. 2-19, shows the constellation diagram of BPSK modulated signal.

$$s_{SPK}(t) = \begin{cases} Asin(2\pi f_c t) & , \ binary\ 1 \\ -Asin(2\pi f_c t) & , \ binary\ 0 \end{cases} \quad (2\text{-}1)$$

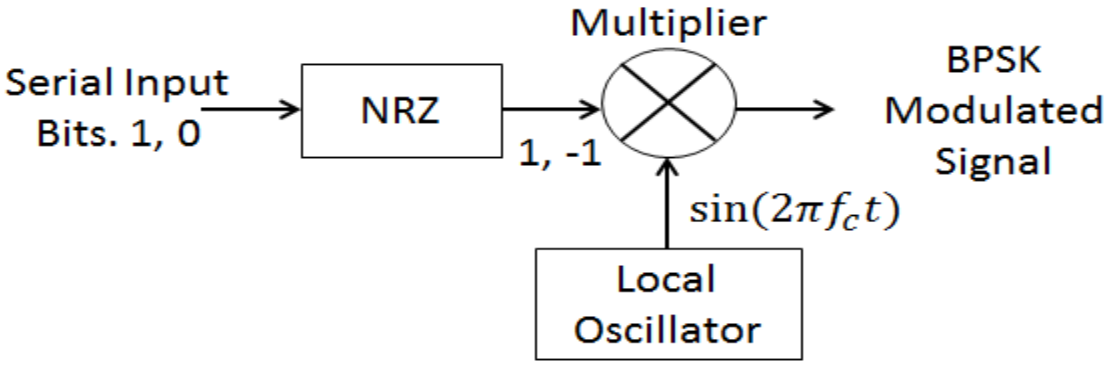


Figure 2-17. BPSK Modulation Block Diagram.

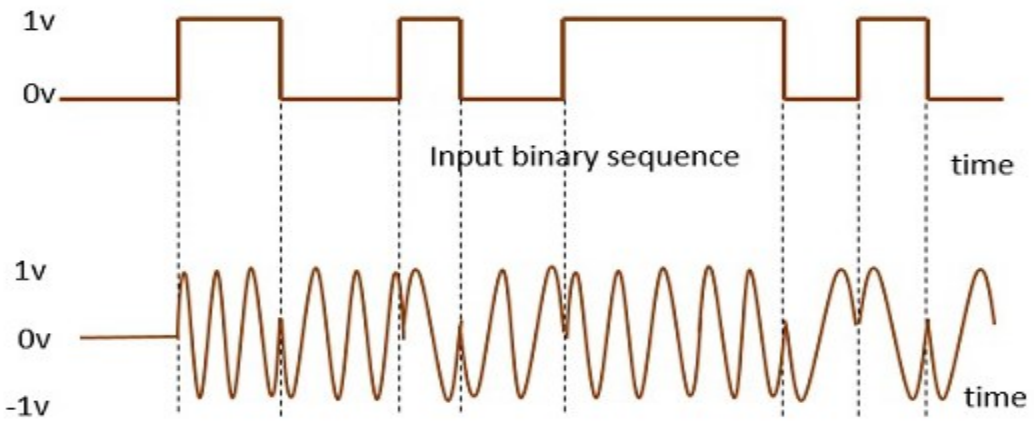


Figure 2-18. Input Binary Data and Corresponding Output Symbols.

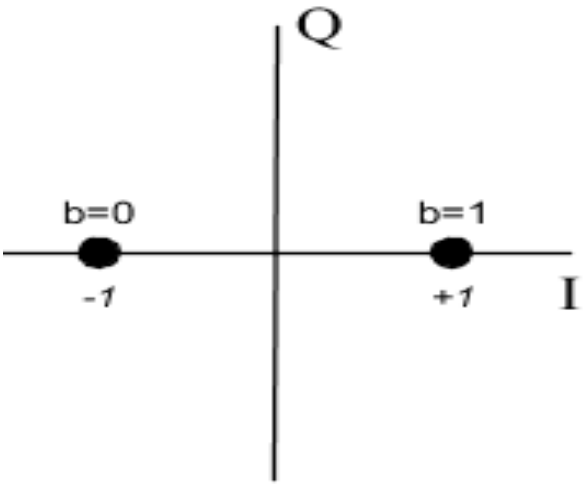


Figure 2-19. BPSK Constellation Diagram.

The baud rate is related to the bit rate, bit rate is the rate of change of the binary input data, and baud rate is the rate of change of the output symbols. For BPSK, bit rate is equal to baud rate, because one output symbol will be generated for every change in the input data. The bandwidth of BPSK modulation is equal to baud rate. Eqs. (2-2, 2-3, and 2-4) expressed calculation of bit rate, baud rate, and bandwidth (BW) assuming input bit period is T_b.

$$Bit\ rate = \frac{1}{T_b} \quad (2\text{-}2)$$

$$Symbol\ rate = \frac{1}{T_b} \quad (2\text{-}3)$$

$$BW = \frac{1}{T_b} \quad (2\text{-}4)$$

2.4.1.2. BPSK Demodulation

To demodulate the received signal, first the carrier of the received signal has to be recovered using Costas loop or phased Locked Loop (PLL). After that, the recovered carrier signal is multiplied by the BPSK modulated signal then the results are integrated and passe through level detector to compare the input value with a threshold to determine and regenerate the corresponding binary data as seen in Fig. 2-20.

A bit clock rate is needed at the detector circuit to produce the original binary message signal. If the bit rate is a sub-multiple of the carrier frequency, then the bit clock regeneration is simplified [30].

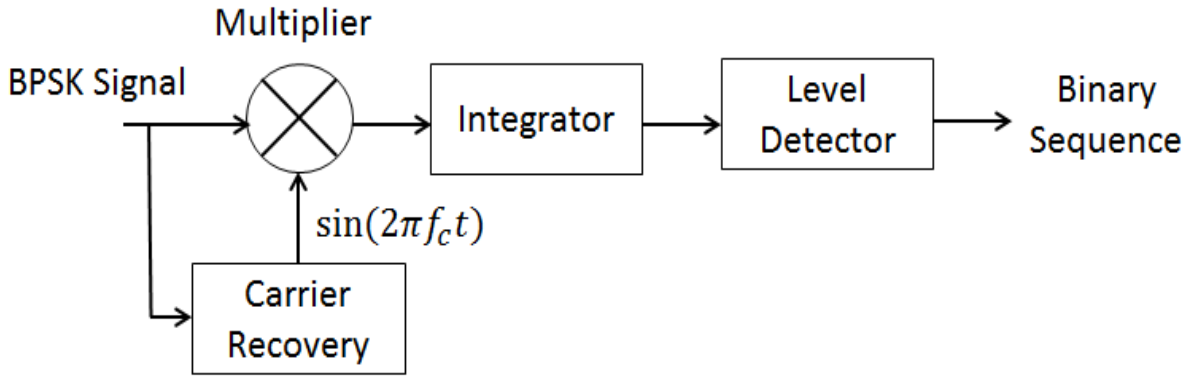


Figure 2-20. BPSK Demodulation Block Diagram.

2.4.2. QPSK Concept

In QPSK modulation, the input binary data required to be transmitted through transmission medium are represented by four cosinewave symbols of phases 90º, 135º,

225º, and 315º. This section describes both modulation and demodulation processes overview of QPSK modulation type.

2.4.2.1. QPSK Modulation

QPSK can be called a di-bit system where the number of changes in the two successive bits of the input steam is equal to four changes which lead to have four output symbols to be transmitted. Each symbol in QPSK has a distinct value of the carrier phase shift. QPSK signal has high data rate in the same bandwidth than that of a single-bit system because QPSK can carry twice as much data as a single-bit system can, the provided SNR is high enough.

QPSK has four different phase shifts and is separated by multiples of 90º of the carrier signal C(t) as shown in Eq. (2-5) where A shows the amplitude and f_c is the carrier frequency. For a single carrier frequency, there are possible four output symbols (phases), corresponding to I and Q bits; 00, 01, 10 and 11 di-bits. Each di-bit generates one of the four possible output phases (45º, 135º, 225º, and 315 º) as in Eq. (2-6) [31]. The rate of change at the input (bit rate) is equal to twice the change at the output (baud rate).

$$C(t) = A2\pi f_c t \tag{2-5}$$

$$s_{QPSK}(t) = \begin{cases} Acos\left(2\pi f_c t + {}^{\pi}/_{4}\right) & , \ binary\ 11 \\ Acos\left(2\pi f_c t + {}^{3\pi}/_{4}\right) & , \ binary\ 01 \\ Acos\left(2\pi f_c t - {}^{3\pi}/_{4}\right) & , \ binary\ 00 \\ Acos\left(2\pi f_c t - {}^{\pi}/_{4}\right) & , \ binary\ 10 \end{cases} \tag{2-6}$$

The generation processes of QPSK modulated signal start with the conversion on the binary input data to parallel or di-bits data where these di-bits are divided into even and

odd bits. Even bits represent I-channel (in-phase) and odd bits represent Q-channel (quadrature-phase). I-channel bits are multiplied by the carrier signal and the Q-channel bits are multiplied by the carrier signal but with a 90° phase shift. The multiplication results are combined to generate QPSK modulated signal as seen in Fig. 2-21 [32].

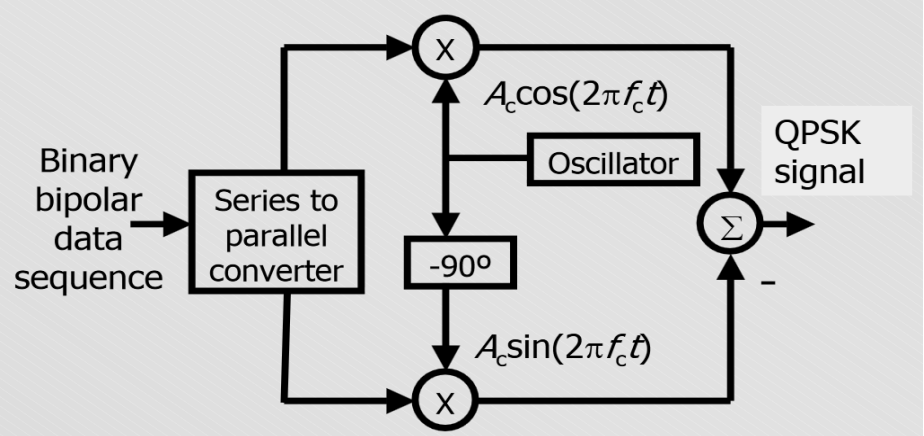


Figure 2-21. QPSK Modulation Block Diagram.

Fig. 2-22 illustrates the generation processes of QPSK modulation, i.e., the corresponding output symbol according to di-bits input binary data. Fig. 2-23 shows the constellation diagram of QPSK modulated signal.

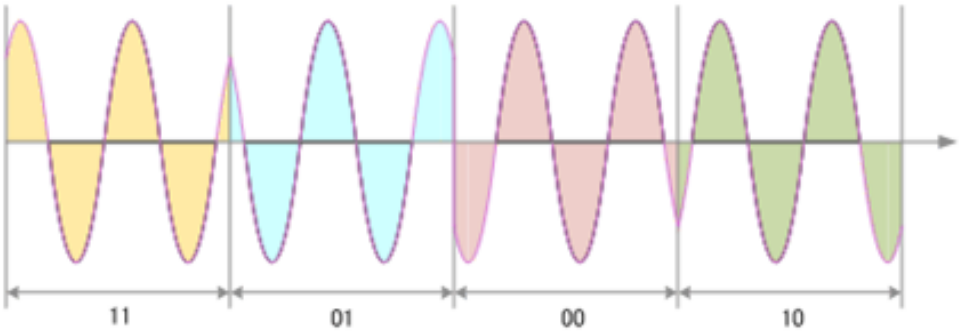


Figure 2-22. Input di-bits and Corresponding Output Symbols.

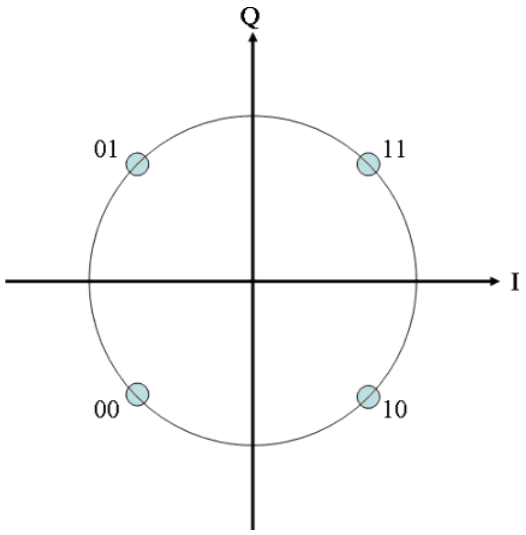


Figure 2-23. QPSK Constellation Diagram.

The baud rate is related to the bit rate, bit rate is the rate of change of the binary input data, and baud rate is the rate of change of the output symbols. For QPSK, bit rate is twice baud rate, because one output symbol will be generated for every change in di-bit. The bandwidth of QPSK modulation is equal to baud rate. Eqs. (2-7, 2-8, and 2-9) expressed calculation of bit rate, baud rate, and bandwidth (BW) assuming input bit period is T_b.

$$Bit\ rate = \frac{1}{T_b} \tag{2-7}$$

$$Symbol\ rate = \frac{1}{2T_b} \tag{2-8}$$

$$BW = \frac{1}{2T_b} \tag{2-9}$$

2.4.2.2. QPSK Demodulation

To demodulate the QPSK modulated received signal, first the carrier of the received signal has to be recovered using Costas loop or PLL. After that, the received signal passes through the upper and lower branches of the demodulator, and then the recovered signal is multiplied by carrier signal in I-channel and multiplied by the carrier signal but with a 90^o phase shift in Q-channel, Consequently the results from both branches are integrated and pass through level detector which compares the input value with a

threshold value to determine and regenerate the corresponding binary data. These binary data are multiplexed to regenerate the binary sequence as seen in Fig. 2-24.

A bit clock rate is needed at the detector circuit to produce the original binary message signal. If the bit rate is a sub-multiple of the carrier frequency, then the bit clock regeneration is simplified [33].

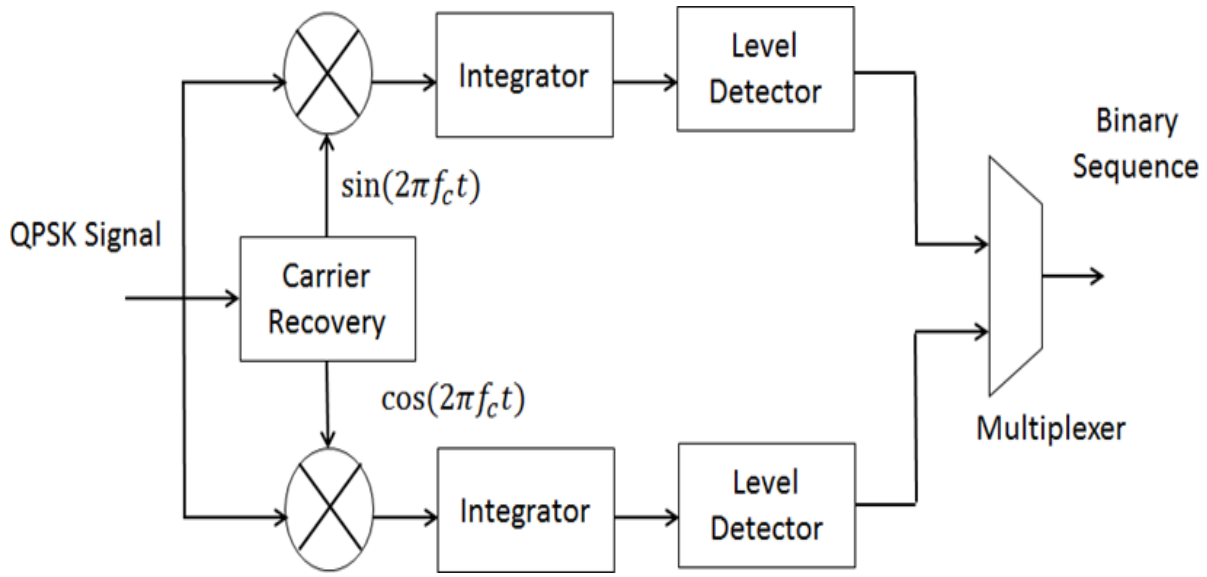


Figure 2-24. QPSK Demodulation Block Diagram.

2.5. Related Work

Because of the significant role that ADMR plays in Cognitive Radio which aims to identify the modulation technique of the received signal, many researches have implemented to enhance the capabilities of such systems. The combination and merging among ADMR and in-flight reconfigurability, and dynamic partial reconfiguration of FPGA increase the hardware flexibility, programmability, and adaptive functionality during runtime, which improve the performance and maintenance of space applications.

In [34], ADMR was implemented using Stockwell transform (S-transform). S-transform is a feature extractor method that extracts distinctive features (energy and entropy) of

several modulation techniques as BPSK, QPSK, FSK and MSK, and allows localization of the signal into time and frequency domains. The extracted features are classified by using different classifiers such as Support Vector Machine (SVM), Artificial Neural Network (ANN), Linear Discriminant Analysis (LDA), k-Nearest Neighbor (k-NN), and Naive Bayes (NB). This system is tested with the presence of Additive White Gaussian Noise (AWGN) of Signal to Noise Ratio (SNR) varying from 0 to 20 dB and the results the high recognition rate with good classification accuracy and low computational complexity.

ADMR plays important roles in different application area especially in Software Defined Radio (SDR), threat, and surveillance analysis. ADMR system can be implemented using Xilinx Virtex-4 FPGA for advanced communication payload. This system based on wavelet transform as a features extractor and statistical calculation of mean computation, and decision threshold logic for the classification method where threshold value is obtained from MATLAB/Simulink. The system was tested using Quadrature Amplitude Modulation (QAM), PSK and FSK digital modulation signal with the presence of AWGN and had high performance [35].

In [36], the detection of the received signal and the identification of their modulation technique could be realized by using spectrum sensing. The detection of primary user signal could be achieved by using improved energy detection technique and the identification of the modulation type (AM, FM, FSK) could be realized by using a combination of Principle Component Analysis (PCA) for features extraction, and ANN for classification process. Real-world signal was used to evaluate the system performance and the evaluation proved that the modulation type had not any effect on the detection performance wherein the number of samples (N) and the detection probability (p_d) increased, and also had proved that reduction in the dimension of received signal did not influence the classification rate as seen in Fig. 2-25, and 2-26.

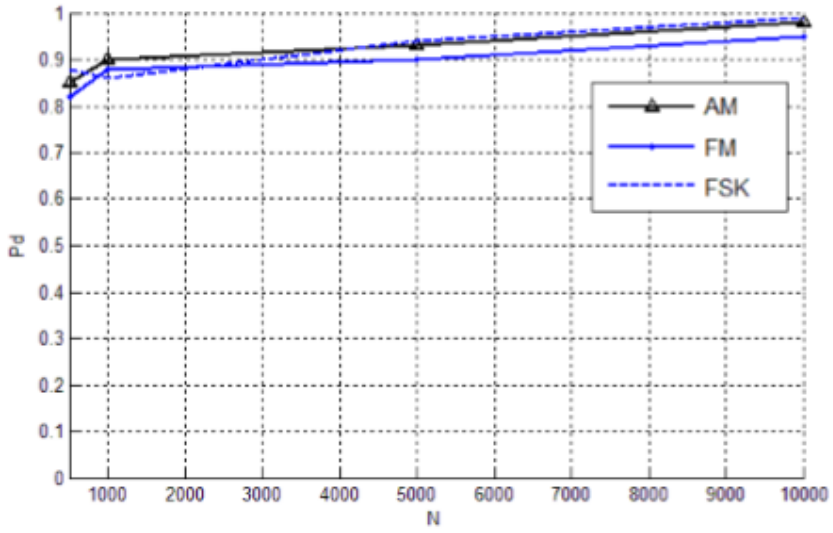


Figure 2-25. Detection Probability According to Number of Samples.

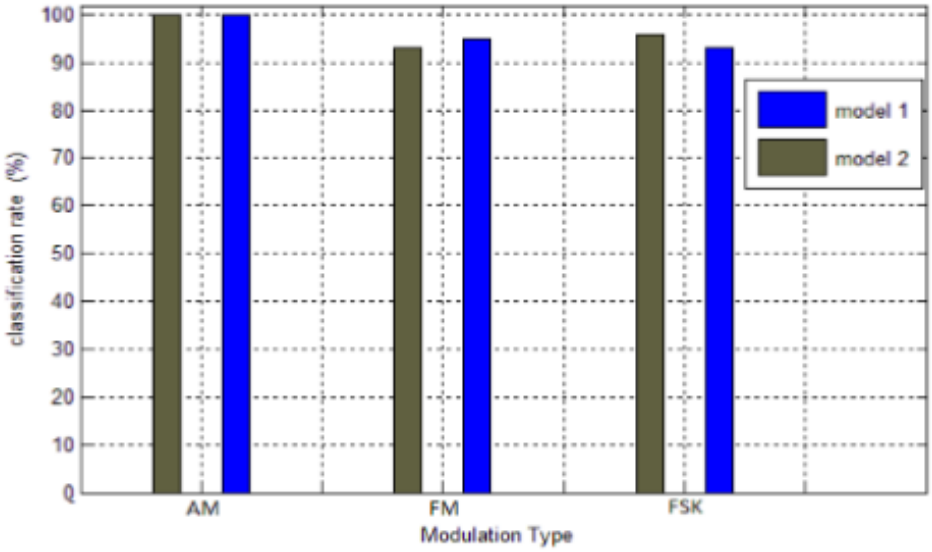


Figure 2-26. Classification Rate for Original and Reduced Data.

Based on the transition of the data symbol of signal stream could be used as unique features templates to represent the features of digital modulation. These transitions could be a change in frequency, amplitude, or phase of digital modulated signal. BASK, BFSK, and BPSK modulation schemes were used to evaluate the system, and Daubechies 1 (Haar) wavelet was used as feature extractor and to construct the templates. The

classification was realized by comparing the matched templates with the extracted features. The results showed recognition rates of 94.8%, 82.5%, and 97% at SNR=-5 dB for BPSK, BASK, and BFSK respectively [37].

8PSK demodulator could be used as features extractor of MPSK modulation schemes (QPSK, OQPSK, π/4 DQOSK and 8PSK) symbols. The output from 8PSK demodulator passes through different finite state machine and transition states according to each modulation scheme. Then these features were applied on a bayes classifier to identify the modulation scheme of the received signal. To evaluate the performance of the proposed system, the recognition rate was computed with different number of observed symbols with the presence of SNR. The results showed that the system offered more accurate classification compared to previous methods especially for QPSK and in low SNR values [38].

A combination of cumulants as features extractor, and Multi-Layer Perceptron (MLP) ANN as classifier enhanced the performance of modulation classification systems for satellite multi-receiver system. The objective of this combination was to find at what level (signal, feature, or decision) where exactly the classification decision was more accurate. 3 receivers were used for the evaluation and the results showed the classification result was more accurate at signal level comparing to feature and decision levels with the presence of SNR range from -5 to 10 dB. Classification decision obtained from signal level had recognition rate of 60% and 95% at SNR= -5 and 0 dB respectively [39].

Features extracted from the instantaneous phase and frequency could be used as input to decision tree classifier to obtain a high classification rate of modulation techniques. This system was applied on Xilinx Virtex-4 LX100 FPGA and utilized only 16% of the available resources which gave enough space to implement more signal processing blocks. The features were extracted from median filters and by computing amplitude variance normalized by the squared mean amplitude [40].

Two stages of classification stage were used to classify among BPSK, QPSK, 16QAM and 64QAM modulations. The input to these stages was the cumulants features of the modulated signal. The classifier uses both Genetic Programming (GP) with k-Nearest Neighbor (k-NN). The first stage was used to classify BPSK and QPSK modulations, and the second stage classified 16QAM and 64QAM modulations. The results showed the recognition rate using 1,024 samples of each modulation technique with a number of trails of 10,000 is 100%, 96.7%, 81.18%, and 79.26% for BPSK, QPSK, 16QAM and 64QAM respectively [41].

DPR provides FPGA designs with adaptive functionality during runtime which improve the performance and at the same time can be considered as a good choice for design maintenance. DPR allows the reduction in resources usage since it is not mandatory to build complex design and run it as all even some parts of the design work and special conditions. Furthermore, DPR allows fetching these parts of the design from external storage to reconfigure FPGA with when it is required.

To reconfigure a Partial Reconfigurable Module (PRM) in-flight, it is better to store PRMs into a safe reprogrammable configuration memory with redundant option. FPGAs devices can be affected with ionizing radiation which causes SEU or a value change configuration bits that may affect design performance. Applying DPR on Triple Modular Redundancy (TMR) with the combination of scrubbing can be a smart solution to overcome SEU especially this solution to guarantee the design performance, increase flexibility, save resources and power consumption. The framework of FPGA reconfigurability is illustrated in Fig. 2-27 [42].

Implementing a FPGA based Digital Signal Modulator (DSM) using FPGA Partial Reconfiguration feature has lower cost than dedicated ASIC based DSM. Only one modulation technique is loaded during any point of time which reduces the usage of FPGA resources and overcomes the limited resources of some FPGA devices. This system uses difference-based partial reconfiguration and is implemented on Xilinx

x3s400-5pq208 FPGA device and tested by using two modulation techniques as a prototype QAM and QPSK modulators. The results showed the ability of the system to realize several DSM on FPGA of limited resources [43].

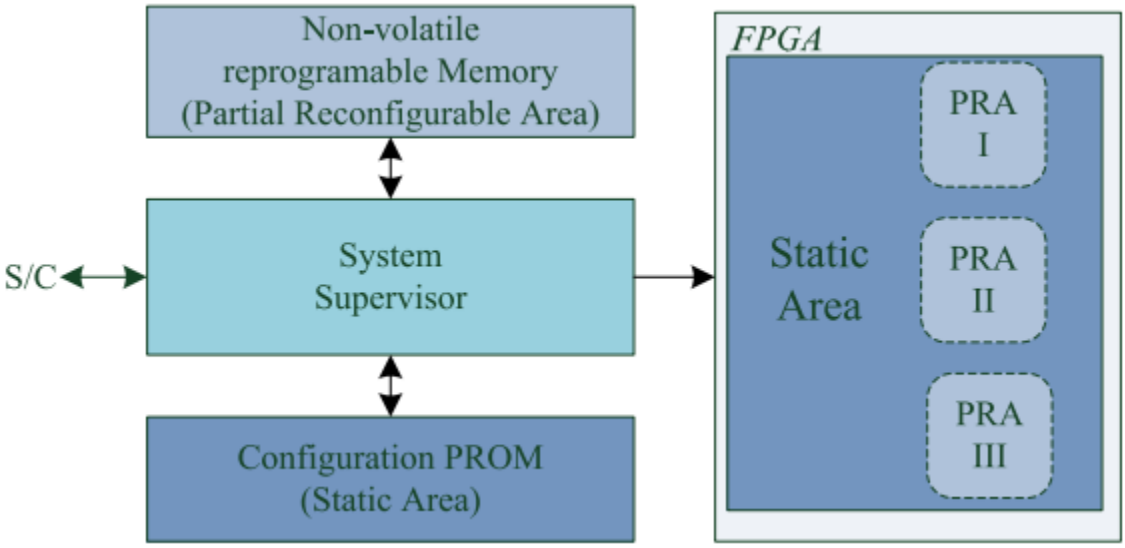


Figure 2-27. FPGA Reconfigurability Framework.

DPR contributes in the implementation of adaptive receiver for telecommunication system such the Orthogonal Frequency Division Multiplexing (OFDM) system based cognitive radio. Following to the input SNR to a configuration controller one of the modulation schemes is implemented and transmitted through a transmitter. Peak to Average Power Ratio (PAPR) is analyzed and observed for each modulation scheme. Results prove that using DPR reduces hardware cost and configuration time and increases flexibility by using reconfigurable modules [44].

Implementation of a new SDR platform can be realized based on DPR. This new platform implements several modulation techniques by using COTS component. The platform provides the configurability of different modulation techniques during runtime without additional hardware overhead. The platform is compared to spatial multiplexing Multiple Input Multiple Output (MIMO) systems. The comparison includes power consumption, cost, and size according to a number of transmitting antennas and spectral

efficiency and the platform. The results show advantages of the platform over the spatial multiplexing MIMO systems [45].

In chapter 3, implementation of the classification part of DPRDS design and evaluation system on FPGA are described in details as well as features extraction method, SVM classifier FPGA implementation, and BPSK and QPSK modulation and demodulation implementation based on SDR principle.

Chapter 3 : Dynamic Partial Reconfigurable Demodulation System - Classification

Increasing the demands of using satellites especially small satellites and its rapid growth rate (see Fig. 3-1) in various applications such as Earth observation, communication, scientific research, and technology demonstration have motivated satellite designers to make the best use of the available size of a satellite by reducing the hardware complexity and at the same time to reduce the total power consumption and cost. One of the solutions is the implementation of hardware function by the means of software using SDR principle.

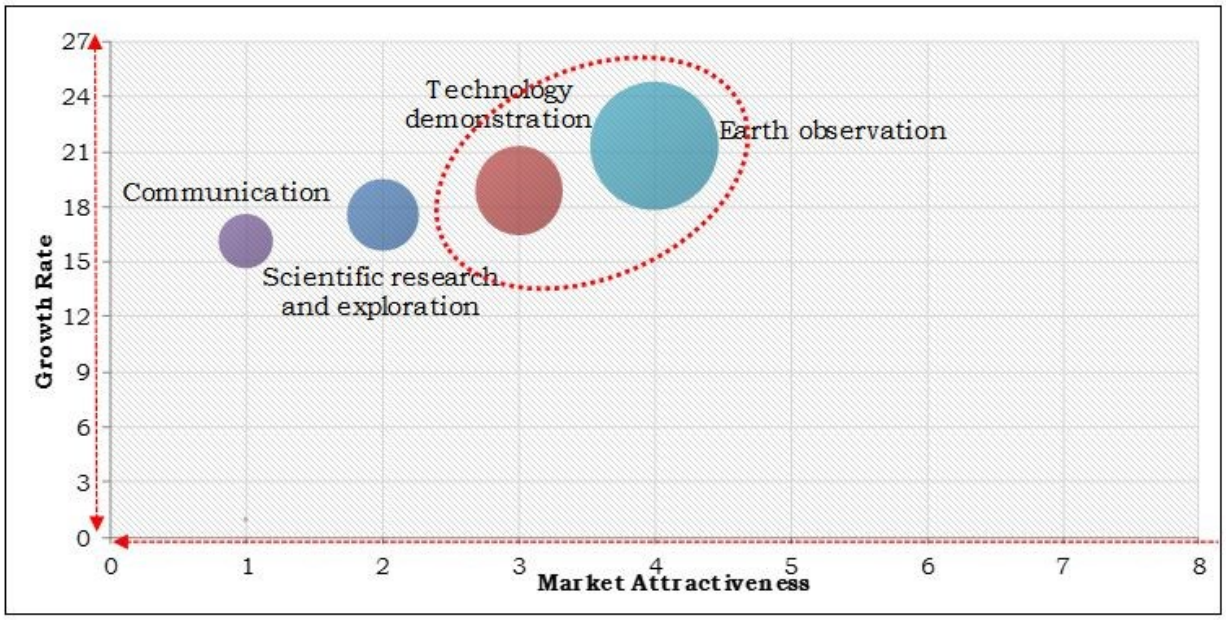


Figure 3-1. Growth Potential of the Global Small Satellite Market

So, to increase the capabilities of a satellite to communicate with several GCSs which use different modulation techniques without installing several receivers into a satellite. On the other hand, the most popular space modulation techniques can be all implemented by software using SDR principle. In that case, an automatic modulation classification

function must be implemented to identify the modulation type for responding to the party at the other end with the proper demodulator.

This chapter describes the implementation of Automatic Digital Modulation Classification (ADMC) or the classification part of Dynamic Partial Reconfigurable Demodulation System (DPRDS). DPRDS consists of two parts; ADMC or classification and Dynamic Partial Reconfiguration (DPR) parts. DPR part is described in chapter 4. In addition, the evaluation method of DPRDS is described to evaluate the performance of the system and to show the real output results after implementing the system on FPGA.

ADMC is segmented into two segments; features extraction and classifier segments. Discrete Wavelet Transform (DWT) is used in this dissertation as features extraction method which extracts the features of the received modulated signal because it represents the modulated signal in time and frequency domains. Support Vector Machine (SVM) classifier uses Radial Base Function as a kernel function to classify between MPSK modulation schemes; BPSK and QPSK with high accuracy. SVM classifier has two stages; one for training (offline) and the other for classification (online).

3.1. DPRDS – Classification

According to the purpose of the research, a modification on the simplified pattern recognition system shown in Fig 2-6 must be implemented. A feature extraction module must be added to extract received signal features before applying this signal to a demodulator as shown in Fig. 3-2. ADMC part of DPRDS is implemented based on this modification as seen in Fig. 3-3.

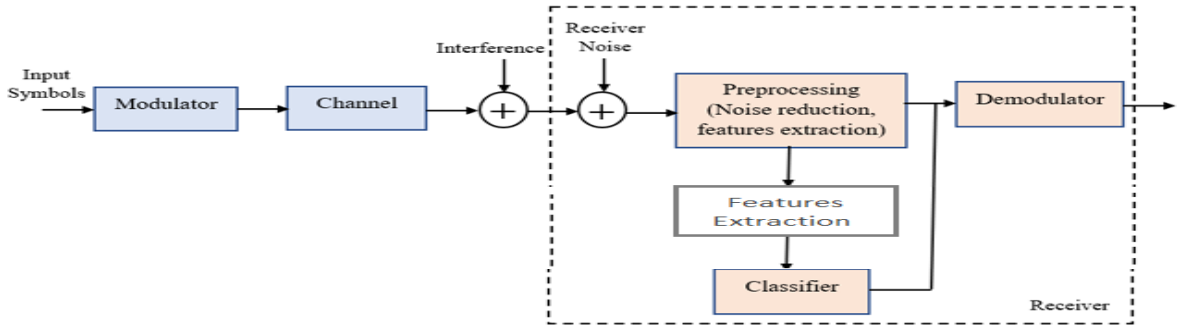


Figure 3-2. Modified Basic Pattern Recognition System.

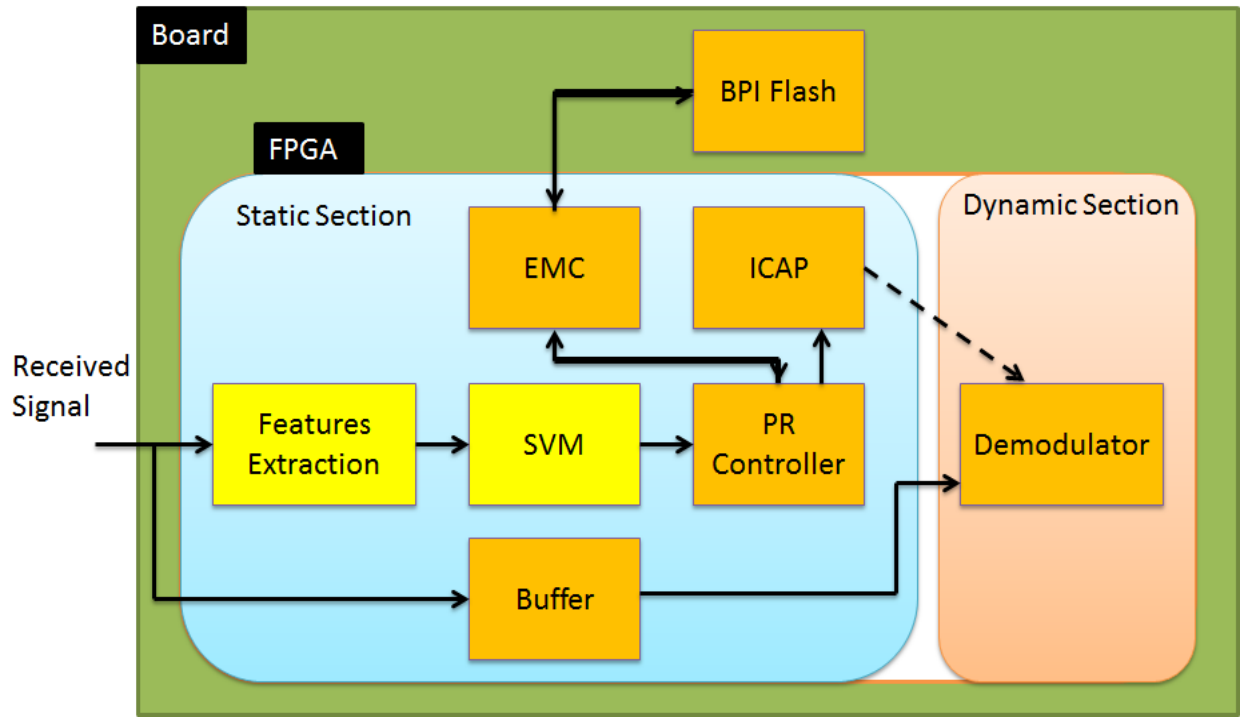


Figure 3-3. DPRDS Design Block Diagram.

In Fig. 3-3, the block diagram of DPRDS design is illustrated. The DPRDS design is divided into two sections; static and dynamic sections. The static section contains the part of the design that is fixed and cannot be changed during runtime, and dynamic section contains the part of the design that can be changed during runtime without any degradation of the static section performance.

The static section contains of two modules sets; one modules set is responsible of identifying and recognizing the modulation type of the received modulated signal (ADMC), and this set has two modules; features extraction and SVM classifier modules. The other set is responsible of performing the dynamic partial reconfiguration process, and it is discussed in details in chapter 4.

In this chapter, the set that responsible of identifying the modulation type is described. This set consists of features extraction and SVM classifier modules which are highlighted on yellow in Fig. 3-3.

The meaning of the word features is that any extractable measurements represent the input signal. The benefits of extract features from input signal and not to manipulate directly with the signal itself (raw input signal) reduces the dimension of the measurement by extracting the distinctive features, which should be robust, simple, stable, and fast to calculate. Reducing the dimension of the measurements reduces the architecture complexity of the SVM classifier. The features extractor of DWT converts the received digital modulation signal into a series of numerical descriptors, called feature vectors.

The classification of the modulation type of the received signal during runtime required first to train the classifier on BPSK and QPSK modulation schemes features. The performing of modulation identification process has two operation modes; a training and classification operation modes. During training mode called offline mode, the extracted features form both modulation schemes are used to train SVM classifier. After the training process, the training results which contain information and parameters that are required during the classification mode are stored into SVM system model database to be used during the implementation phase of the classifier on FPGA. The training mode is performed by a combination of VIVADO and MATLAB tools as will be discussed later in this chapter.

During classification mode or runtime mode, features from the received modulated signal are extracted, which has unknown modulation type for the SVM classifier, after passing through DWT features extractor module. These features have lower dimensions than the raw input signal. The extracted features are then passed to the classifier. The classifier performs the classification calculations according to the classification equation and using the information and parameters obtained during the training mode. In another meaning, SVM classifier performs pattern matching of the extracted features with the one created into a model during offline mode and used during the implementation of the design on FPGA. According to the results of pattern matching process, SVM classifier takes an identification decision about the modulation type as shown in Fig. 3-4.

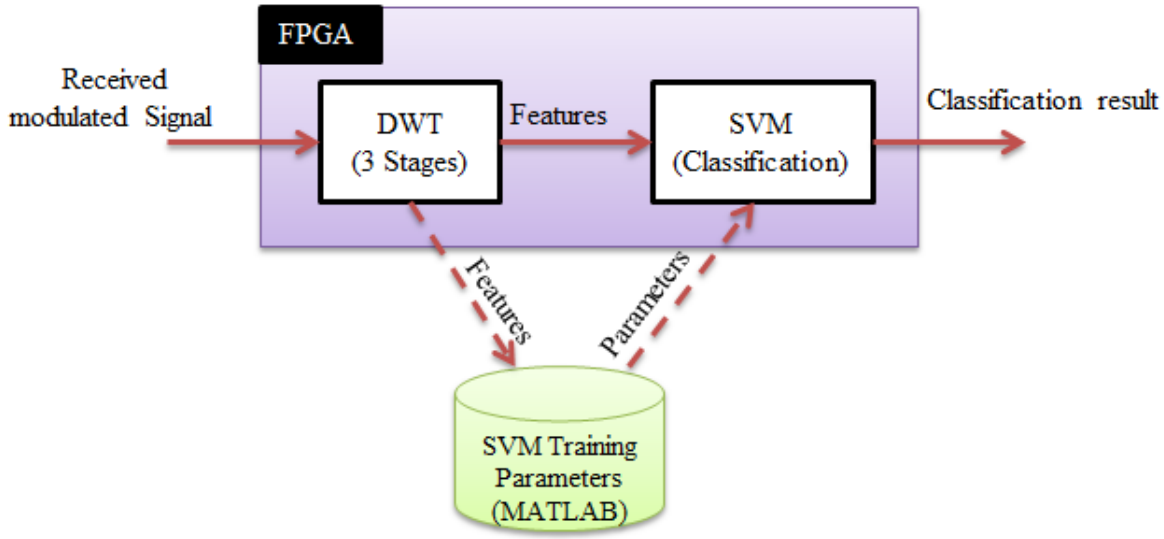


Figure 3-4. Modulation Classification Process.

The process of pattern matching refers to one or several algorithms, that calculate a matching sheet between the features of unknown modulation type vector and the one obtained during training. The pattern matching process output is some of numeric values which are related to a class of the two classes. Thereafter, the classification decision is taken by comparing these results to a threshold and makes the final decision of the modulation type.

Fig. 3-4 illustrates the principle of ADMC part of DPRDS design. It is remarkable that the approach is used for recognition of the modulation type, and the algorithms of both features extraction and the classification processes are of critical importance to recognition system. In the following sections, features extraction method of DWT and its FPGA implementation, in addition to SVM classifier architecture, kernel method, testing and FPGA implementation, also the evaluation system of DPRDS are discussed in detail in this chapter.

3.2. Modulation Type Classification Based on Pattern Recognition

In this dissertation, pattern recognition approach is used to implement the classification part of DPRDS system. The selected features extractor is DWT. The selection of DWT is based on its ability to extract distinctive features form each modulation type which are simple, stable, and fast to calculate. In addition, DWT reduces the dimension of the features which reduces the complexity of the classifier. SVM is used as a classifier of the modulation type according to the output features from DWT stages.

The sequence of the manipulation of the received modulated signal starting from the training stage (offline) till the classification stage (runtime) and how the ADMC part is deal with these features to obtain perfect classification results are illustrated in Fig. 3-5.

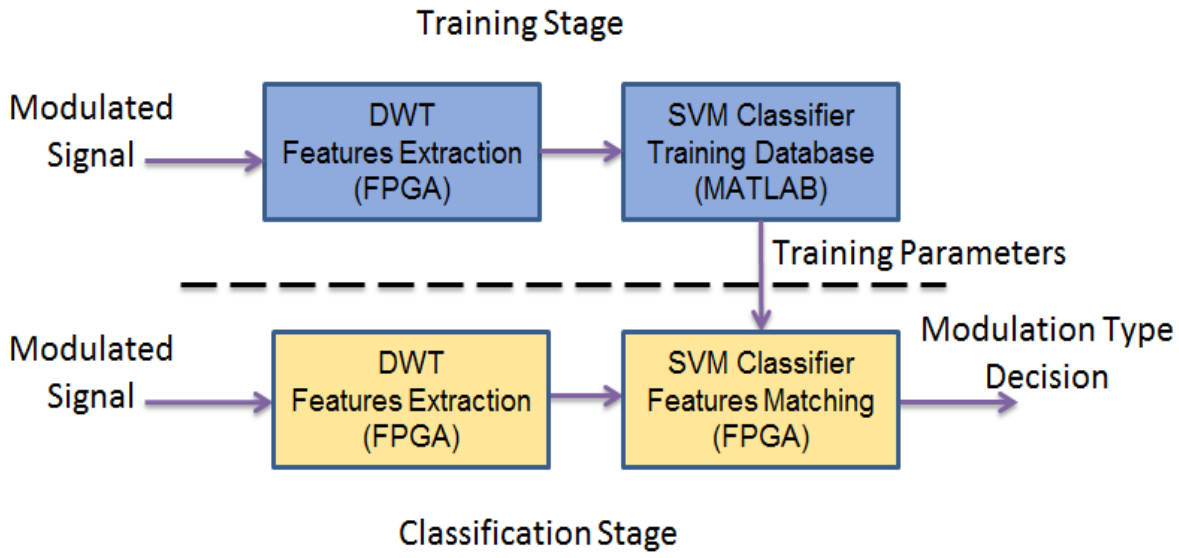


Figure 3-5. ADMC Operation Concept.

3.2.1. FPGA Based DWT

In DWT, the signal transform does not change the information content existing in the signal, and it is just another representation form of the signal. It means that the information content exist in the signal does not change. Wavelet transform allows the localization of the signal in time-frequency domains. i.e., DWT represents the signal with components in both time-frequency domains. The merits of representing a signal in both

time and frequency domains are important for pattern matching operation in two ways. Firstly, different amounts of information may be transferred by different parts of the signal. Secondly, if a local noise corrupts the signal in time and/or frequency domain, the noise affects only a few coefficients because local information is represented by coefficients in the time and frequency domains.

DWT is based on sub-band coding by dividing a received modulated signal into several sub-bands of different scales. Sub-band coding provides the studying of each scale, separately, allows DWT to reduce resources required and the computation time, and to be easy to implement [46].

Filters are one of the most popular signal processing functions. Wavelets can be achieved by repetition of a group of filters with rescaling. The signal resolution is a measure of the amount of detail information in the signal and it is realized by the filtering processes and the rescaling is realized by up-sampling and down-sampling processes.

DWT is computed using a series of low and high pass filters (LPF and HPF), which means it represents a given modulated signal as a series of approximate coefficients obtained from LPFs, and detail coefficients obtained from HPFs at different resolutions. The resulting coefficients from both LPF and HPF are down sampled by 2. Two dimensions (2-D) or two decomposition levels of DWT is shown in Fig. 3-6.

The LPFs and HPFs at each decomposition level remove half of the frequencies with double the frequency resolution where the uncertainty in frequency is reduced by half. According to Nyquist's rule, after removing the half of the frequency band, the signal can be sampled at half of the frequency thus removing half of the samples without any loss of the signal information, i.e., it halves the time resolution of the modulated signal. So, the signal is now represented by half of the samples. The reduction of time resolution is called decimation by 2.

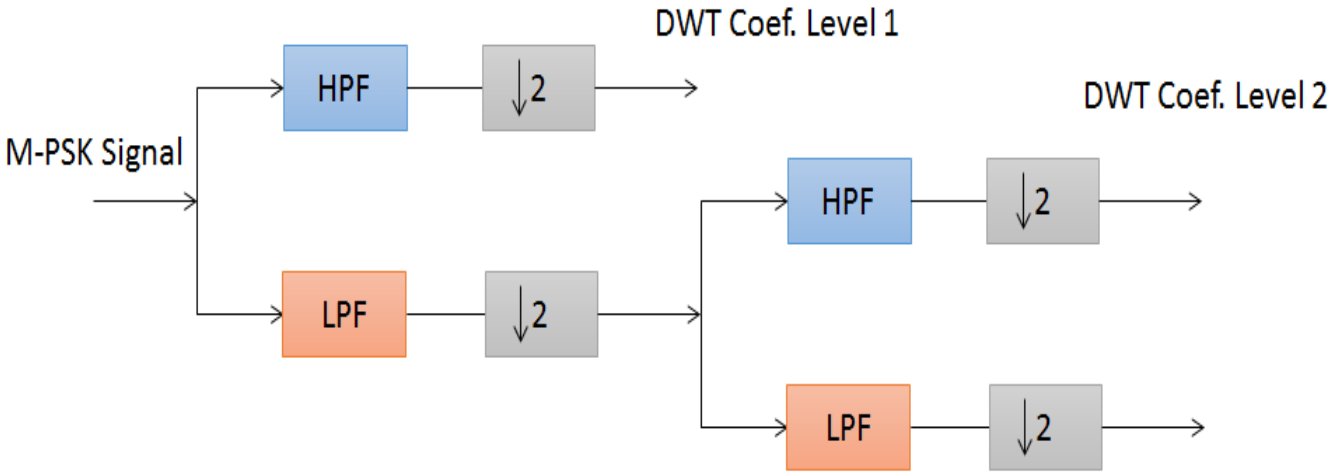


Figure 3-6. 2-D DWT Configuration.

The number of the DWT stages plays an important role in determination of how the extracted features are distinctive for the classifier and the number of required features. The more number of stages, the less number of features is required to represent the modulated signal, so it is important to take into consideration, the fewer number of features will not be enough to support SVM to produces the correct decision, so it is better to tradeoff between the number of stages and the number of features.

In this dissertation, 1, 2, and 3 D of DWT are tried to generate the proper features/ coefficients based on these dimensions. SVM classifier talks the correct classification decision, but only 3-D DWT succeeds to generate the proper coefficients (distinctive features) for the classifier. To configure 3-D DWT, it is required to connect the output from the LPF that generates approximate coefficients to the input of the next stage and so on. The output of HPF at the last stage which generates detail coefficients of the signal is used to model the modulated signal in classification process.

The implementation of 3-D DWT is shown in Fig. 3-7. To realize 3-D DWT, only the colored path is mandatory to be implemented. According to the purpose of this research and to realize a 3-D DWT, it is not mandatory to implement the whole blocks in Fig. 3-7,

the colored path is only mandatory to obtain the realize DWT for generating the DWT level3 coefficients (features). The implementation of complete DWT configuration is mandatory for reconstruction of images for many computer vision purposes. So the modulated signal has to pass through two LPF and one HPF to extract its features or DWT coefficient level3 [47].

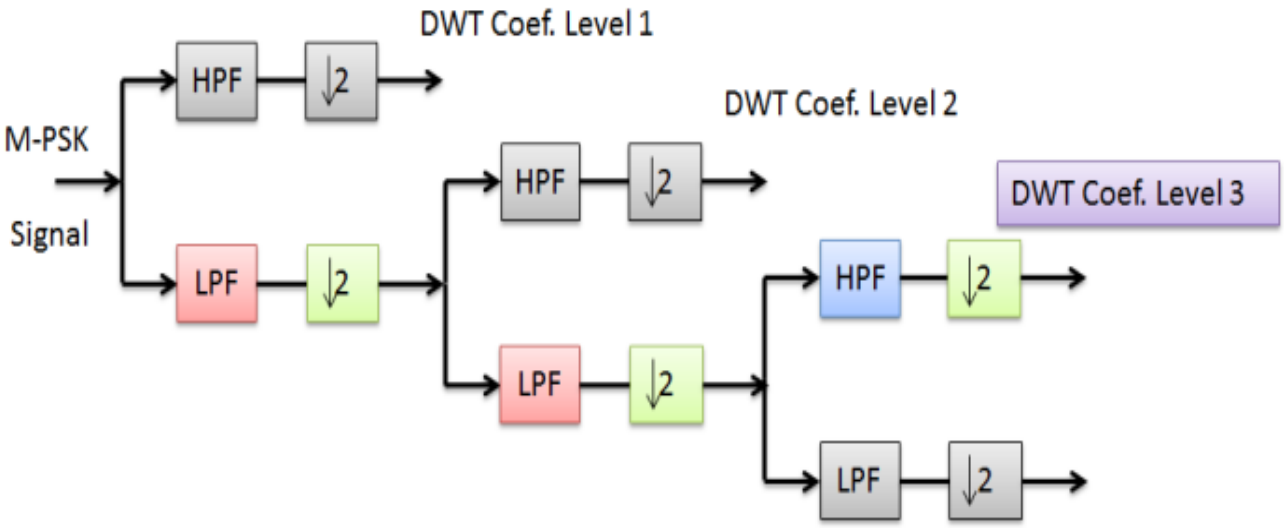


Figure 3-7. 3-D DWT Configuration.

To realize a 3-D DWT on FPGA, three Finite Impulse Response (FIR) IP core filters are used as seen in Fig. 3-8. Two LPF filters followed by one HPF are required to be implemented to obtain the required level of DWT features. Each of these filters is followed by decimation of 2 to half the time resolution and to remove half of the input samples. As mentioned in the previous paragraph, the implementation of whole 3-D DWT structure is mandatory for images reconstruction purposes.

The type of the wavelet used for features extraction is Daubechies 5 wavelet (db5). To obtain the coefficients of this wavelet filter (LPF and HPF coefficients), a MATLAB function ("wfilters") is used to obtain these coefficients. 10 coefficients are used and added during the setup of the three FIR IP core filters. Also the decimation parameter of these filters has to be set to 2 (decimation by 2).

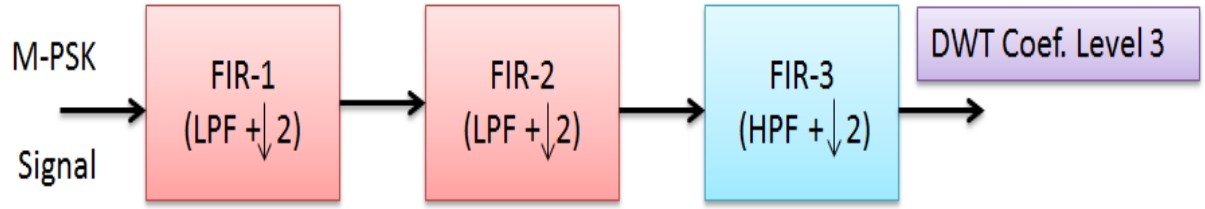


Figure 3-8. Realization of 3-D DWT on FPGA.

3.2.2. FPGA Based SVM

SVM is a binary classifier used for both linear and non-linear classification problems. It is a supervised learning classifier where its kernel is based on supervised learning technique and can be used for both classification and regression processes. In addition, SVM is one of the best supervised learning algorithms [48]. Binary classifier means that SVM is a two classes classifier where the classification is based on widening the margin between the classes and transferring the data into higher dimension space.

SVM can be used for several application especially, forecasting, decision-making, and pattern recognition [49]. In this dissertation, the classification is based on pattern recognition using SVM.

3.2.2.1. Binary Classifier Based SVM

SVM separates two classes based on kernel function used during training phase. It can be applied to separable and non-separable data sets classification. Support Vectors (SVs), weights, and bias are three main parameters that are created during training stage of SVM and are used in classification stage to get high recognition rate. SVs are the closest data points to the hyperplane, weight is the error value resulting from presence of data points in wrong class domain which occurs in the presence of low SNR during ADMC process, and b is a scaler.

To solve non-linear classification problem as in ADMC case, SVM has to transfer the data set from input space, the space of similarity (input space), to features space. This is realized by using a dot product to perform this transformation as shown in Fig. 3-9. This dot product is defined as kernel function.

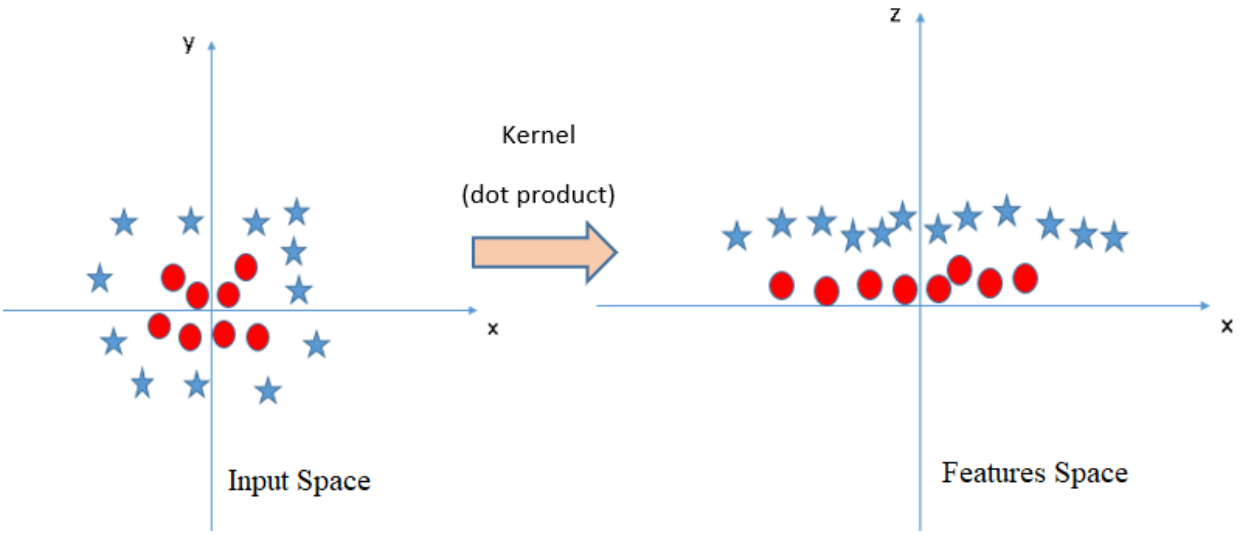


Figure 3-9. Space Transformation using Kernel function.

The main reason of this transformation is that non-linear problem in input space becomes linear operation in the feature space which simplifies classification problem and makes it easy to be solved.

A generalized classifier can be realized by minimizing the training error with obtaining high accuracy for unknown testing set. So, the objectives of SVM are to maximize the margin between the nearest data points and class boundary, and to limit that all data points are belonging to the corresponding class. SVM removes the data points that do not affects these objectives from training data, so the classification decision is based only on SVs and can be achieved using a proper training algorithm [50]. These two objectives are combined into optimization problem to achieve higher accuracy and recognition rate of the modulation techniques.

SVM classifies the data sets belong to two classes into features space by solving optimization function using kernel function that used during training stage. SVM solves the optimization function to find the optimal separating boundary (hyperplane) between these classes. this hyperplane has the maximum distance from the nearest point for each class as shown in Fig. 3-10.

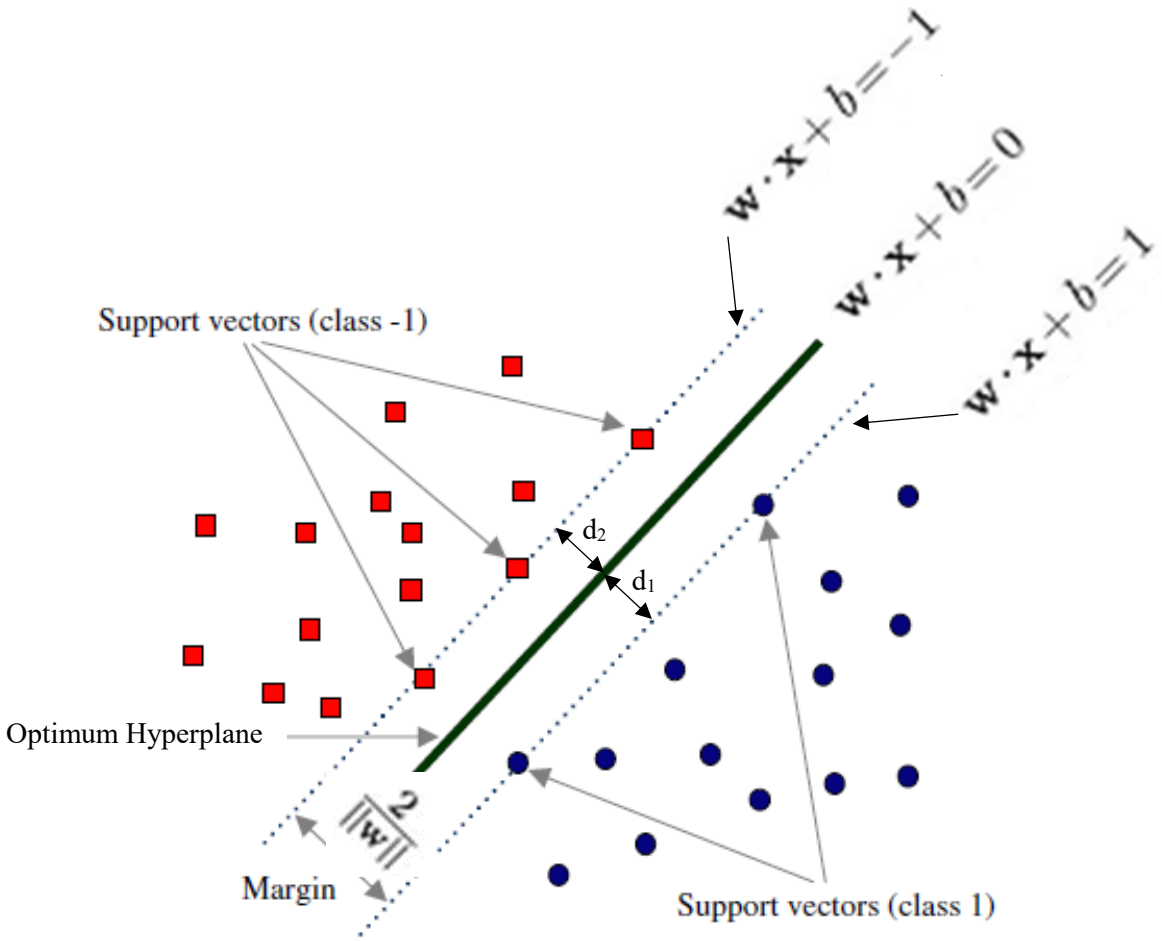


Figure 3-10. SVM Concept.

The margin of a classifier is the distance between two closest data points which belong to each class. The hyperplane is located at the middle of this distance. The simplest SVM classifier is linear SVM (LSVM) classifier and the one that has maximum margin is called maximum margin linear classifier.

For more explanation, assuming a training data set which contains data from each class is represented as $\{x_i, y_i\}$ where i = 1,2,3, .., N, where N is the number of support vectors, and y is the class label ($y \in \{-1,1\}$). the hyperplane that realizes the optimal hyperplane equation (Eq. (3-1)) will be the hyperplane that separates the positive and negative data points of the training example using the classification equation (Eq. (3-2)) [51].

$$w.\ x + b = 0 \tag{3-1}$$

$$f(x) = \text{sign}\ (w\ .\ x + b) \tag{3-2}$$

Where w is the normal to hyperplane and is called weight vector, x is input features vector, and b is the bias as a scaler quantity. Eq. (3-2) can be expressed by another way to show the possible hyperplane (above and below the optimal hyperplane) that separates data points as in Eq. (3-3).

$$f(x) = \text{sign}\ (w\ .\ x + b), \text{ where } sign(x) = \begin{cases} 1, & if\ x > 0 \\ 0, & if\ x = 0 \\ -1, & if\ x < 0 \end{cases} \tag{3-3}$$

Eq. (3-4) constraints that all the training data points exist on either side of hyperplane where the closest points to hyperplane are called SVs.

$$y_i(x_i.w + b) - 1 \geq 0, \forall\ i = 1,2,3\ .... \tag{3-4}$$

Hyperplane can be represented by given (w,b) as w. x + b = 1 deduces the hyperplane for positive class, and w. x + b = -1 deduces the hyperplane for negative class.

The distance between these two hyperplanes is called margin distance and can be expressed as in Eq. (3-5) subject to constraint in Eq. (3-6) [52].

$$Margin = \begin{cases} \frac{2}{\|w\|}, & \text{to maximize } margin \\ \frac{\|w\|^2}{2}, & \text{to minimize } margin \end{cases} \tag{3-5}$$

$$y_i(x_i.w + b) \geq 1, \forall\, i = 1,2,3 \ldots \quad (3\text{-}6)$$

Where the $\frac{b}{\|w\|}$ is the perpendicular distance from hyperplane to the origin, $\frac{2}{\|w\|}$ is equal to the summation of both distances d_1 and d_2. $d_1 = \frac{1}{\|w\|}$, $d_2 = \frac{1}{\|w\|}$ as each positive and negative hyperplanes are located at the same distance from the optimal hyperplane. It is clear that if it is required to increase the margin, so $\|w\|$ has to be minimized.

the received modulated signal contains noise but the effects of this noise can be minimized by definition of a soft margin with marginal error ε which is called positive slack variable as in Eq. (3-7) and subjects to Eq. (3-8).

$$\min_{w,b} \left\{ \frac{\|w\|^2}{2} + C \sum_{i=0}^{l} \varepsilon_i \right\} , \qquad \varepsilon_i \geq 0 \quad (3\text{-}7)$$

$$y_i(x_i.w + b) \geq 1 - \varepsilon_i, \quad \forall\, i = 1,2,3 \ldots \quad (3\text{-}8)$$

Where C is penalty value which controls the tradeoff between the marginal error and testing error. l is the number of data points.

The Lagrangian multiplier α_i is used to reduce minimum problem which can be obtain according to the Karush Kuhn-Tucker condition. The weight vector (w) can be expresses as in Eq. (3-9).

$$w = \sum_{i=0}^{l} \alpha_i y_i x_i \quad (3\text{-}9)$$

Only the data points x_i whose $\alpha_i > 0$ are considered as support vectors, so the classification equation in Eq. (3-2) can be rewritten as in Eq. (3-10)

$$f(x) = \sum_{i=0}^{l} \alpha_i y_i x_i x + b \quad (3\text{-}10)$$

To apply the transformation from input space (non-linear) to features space (linear), a kernel function will be used to achieve this transformation by replacing dot or inner product in Eq. (3-10). So the final form of classification equation can be expressed as in Eq. (3-11).

$$f(x) = \sum_{i=0}^{l} \alpha_i y_i k(x_i, x) + b \quad (3\text{-}11)$$

Where $k(x_i, x) = \emptyset(x_i).\emptyset(x)$ is the kernel function.

SVM uses supervised learning algorithm for the training stage where SVM is provided with input features and their corresponding classes or the target output. The features are passing through the classifier and the output result, which the SVM calculates based on initial values of SVM parameters, are compared with the target output. If the output matches with the target output, the SVM parameters are not changed. If the output does not match with the target output, then the SVM parameters will be adjusted to ensure that the output results match the target output. So this learning type is called supervised because it looks like learning with a teacher mechanism. Fig. 3-11 illustrates the mechanism of supervised learning algorithm [53].

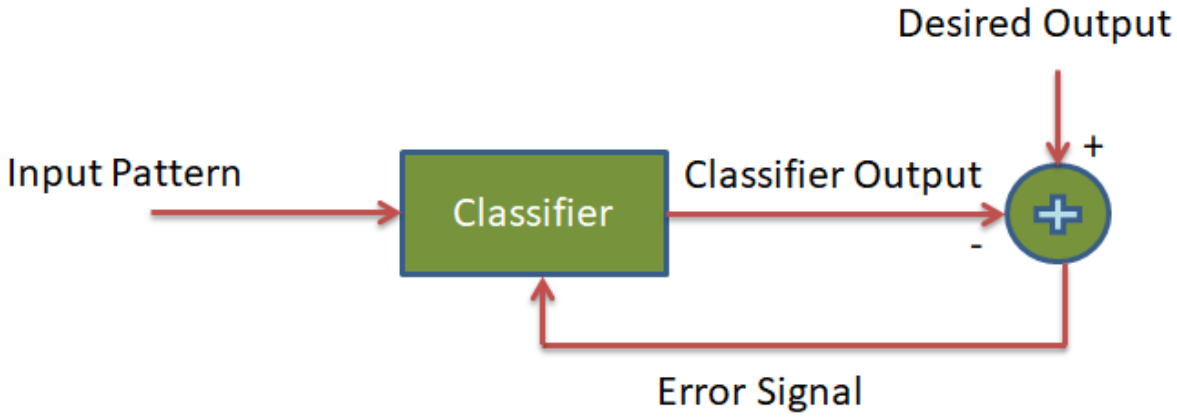


Figure 3-11. SVM Learning Algorithm.

So from Eq.(3-10), and to achieve the classification function on ADMC design on FPGA device, it is required to train SVM classifier (using MATLAB) on the features of the received modulated signal to construct the SVM classifier and to obtain the required

parameters that are needed to build the classifier into FPGA to classify the modulated signal in Realtime.

The required parameters to construct and implement SVM classifier into ADMC design on FPGA device are Lagrange multiplier (α_i), SVs (x_i), bias (b), and kernel function parameters.

Implementation of both training and classification stages of SVM classifier on FPGA consumes a lot of resources and therefore increases the power consumption. So, the efficient way to overcome this challenge is to perform the training stage of SVM classifier with the extracted features (by DWT) from modulated signal using MATLAB R2015b tool (offline training). In this training stage, SVM classifier is constructed and all the parameters required to be used in classification stage are acquired. These parameters (α_i, SVs, b, and kernel function parameters) obtained from MATLAB training stage are used to construct and implement the SVM classifier of ADMC part of DPRDS system on FPGA using VHDL code to classify the modulation type of the modulated signal (BPSK /QPSK) in run time. The concept of operation of training and classification stages and ADMC principle is illustrated in Fig. 3-12.

The obtain high recognition rate of both BPSK and QPSK modulation schemes, it is mandatory to train the SVM classifier with the proper kernel function. The proper kernel function is the one that gives high classification accuracy. In this dissertation, a tradeoff among four different kernel functions is performed to select one of them to be used in training and classification processes of SVM classifier. The four kernel functions are linear, quadratic, polynomial, and Radial Base Function (RBF) as shown in Table 3-1. These kernel functions are trained and test with the features that are extracted using DWT stages using “svmtrain” and “svmclassify” functions of MATLAB R2015b Tool.

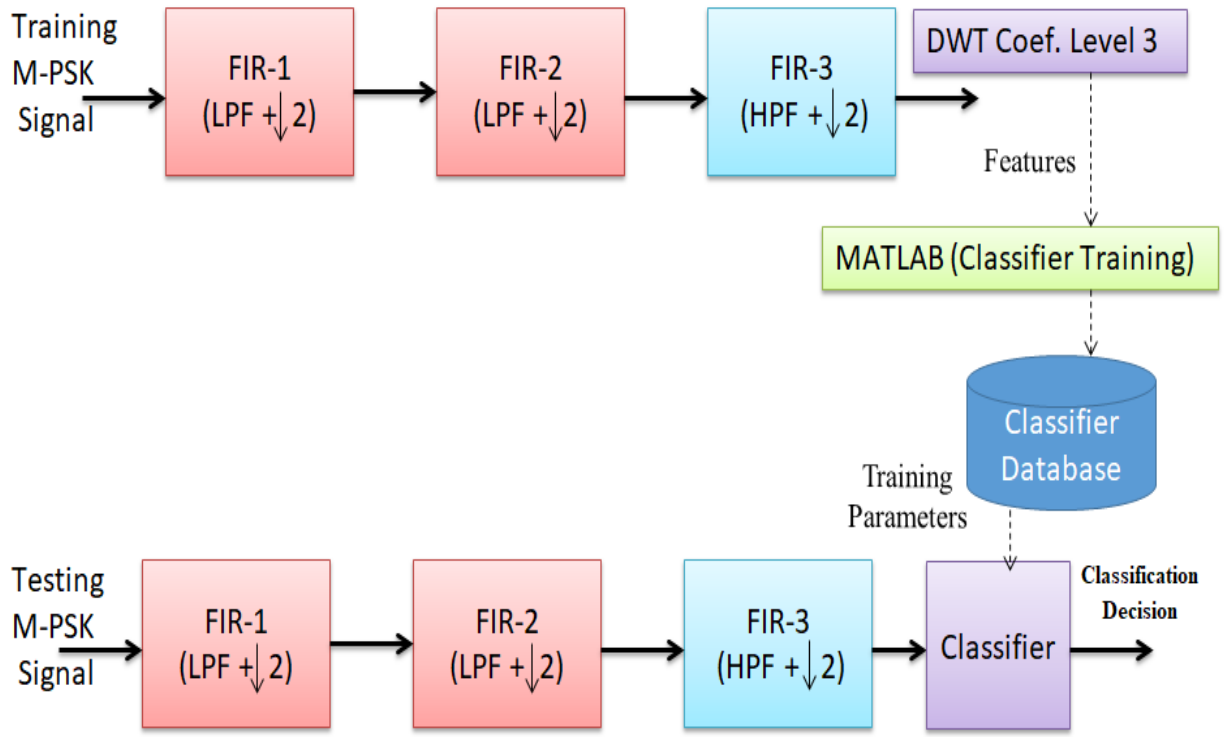


Figure 3-12. ADMC Concept.

Table 3-1. Kernel Functions Mathematical Expression

Kernel Function	Equation
Linear	$k(x_i, x) = x_i . x$
Quadratic	$k(x_i, x) = ((x_i . x) + 1)$
Polynomial	$k(x_i, x) = ((x_i . x) + 1)^{\mathrm{d}}$
Radial Base Function	$k(x_i, x) = exp^{-\frac{\|X_i - x\|^2}{2\sigma^2}}$

Where x_i is SVs vector, x data point vector, d is the degree of the polynomial function, σ is standard deviation and its value determines Gaussian distribution width.

The four kernel functions are tested using the DWT coefficients level 3 features which include 18 features. The selection of the training and testing sets is performed by using cross-validation function. The accuracy of each kernel function is determined by using "classperf" and "correctrate" functions. The comparison among the accuracy of each kernel function after training and testing SVM classifier with the DWT coefficients level 3 features is as shown in Table 3-2.

Table 3-2. Comparison among Different Kernel Function Accuracy

Kernel Function	Accuracy
Linear	**50.50 %**
Quadratic	**54.06 %**
Polynomial	**64.46 %**
Radial Base Function (RBF)	**94.44 %**

Based on this comparison, RBF kernel function is selected to construct SVM classifier in ADMC part of DPRDS on FPGA.

The selection of the value of σ parameter of RBF is determined by trying different values of it (0 to 1 is the suitable range for this application) until high accuracy, so that higher classification percentage can be obtained.

Fig. 3-13 illustrates how the value of σ parameter affects the classification results. The σ parameter value determines the width of SVM classifier boundaries. So, for a larger value of σ, the decision boundaries looks like smooth and flexible but it leads to generate the wrong classification result. For a smaller value of σ, the decision boundary looks like

strict and sharp, comparing to the former situation, and it leads to generate the correct classification result.

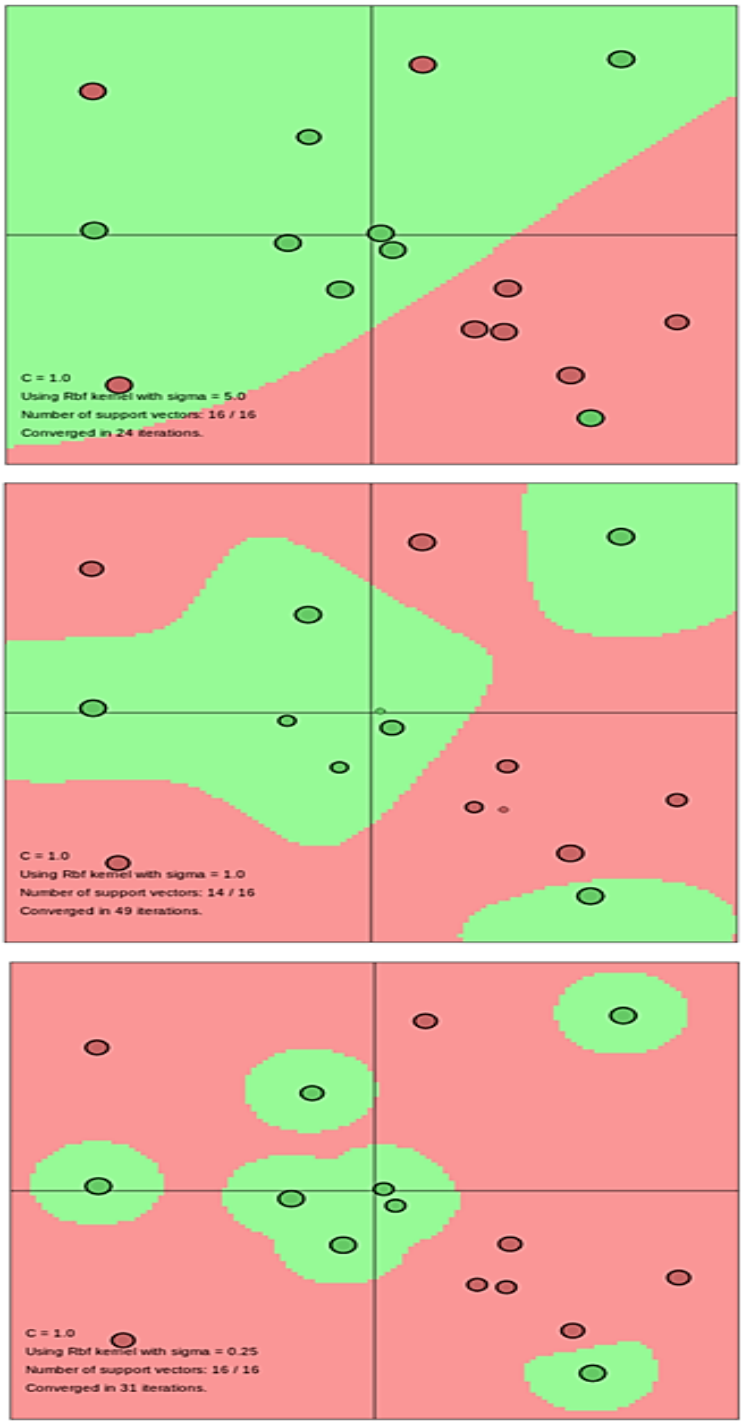


Figure 3-13. Sigma Effects on Classifier Boundaries.

The value of σ parameter for our application is selected to be 0.3 after 18 iterations. This value gives high classification rate for both BPSK and QPSK modulation schemes.

After the selection of the kernel function using MATLAB tool and training SVM classifier on the DWT coefficients level 3 features, the SVM classifier is constructed and the parameters are adjusted to make high classification rate. Therefore, the SVM classifier can be implemented on FPGA using these parameters values according to Eq. (3-10) but the first RBF equation in Table 3-1 has to be implemented. The FPGA implementation of RBF modules of the SVM classifier is illustrated in Fig. 3-14.

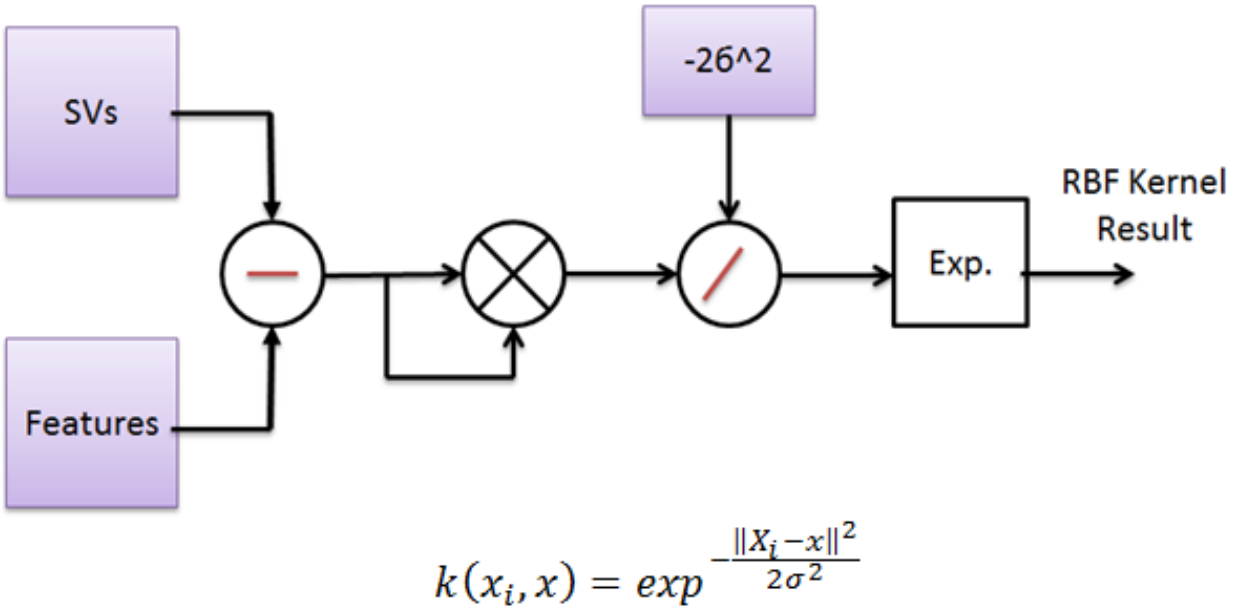
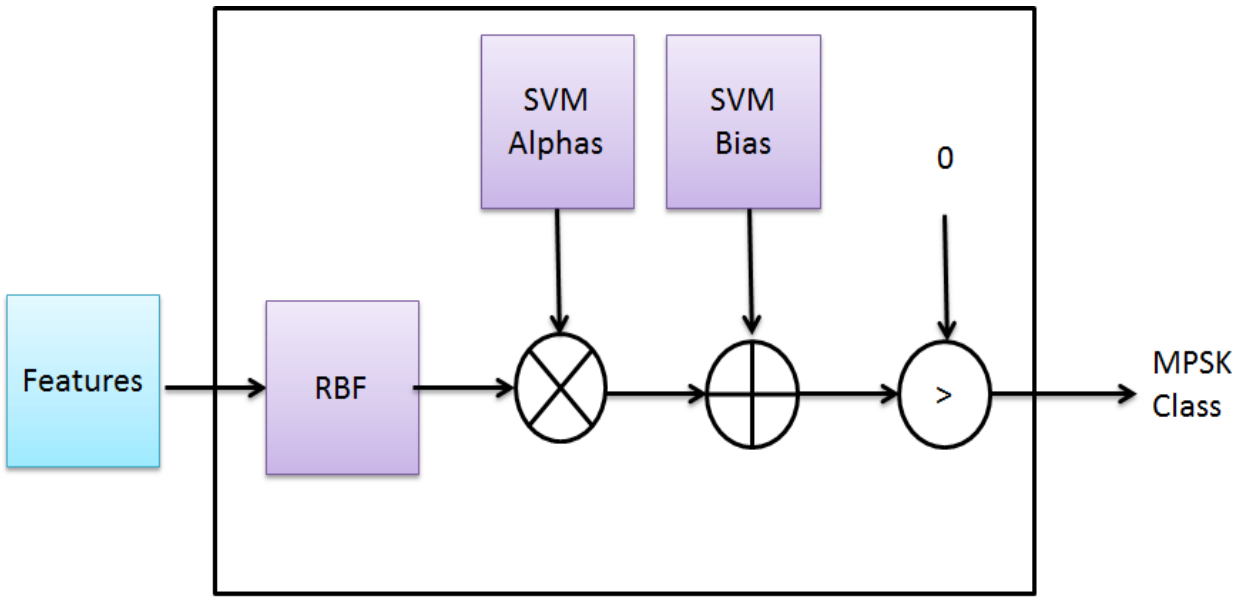


Figure 3-14. RBF Implementation on FPGA.

All the calculation made to compute RBF kernel function, for the SVM classifier, are made by using fixed point numbers to obtain high accuracy. The sequence of RBF function computation is started with the subtraction of input features from SVs, and then the result is squared by multiplying the result by itself and the result of multiplication is divided by $(-2\sigma^2)$. There is no function in fixed point library for performing the

exponential computation process. So, a Look-up Table (LUT) is used to support the exponential calculation.

After implementing RBF on FPGA, the SVM classifier can be implemented on FPGA in the same way the classification equation (Eq. (3-10)) as shown in Fig. 3-15.



$$f(x) = \sum_{i=0}^{l} \alpha_i y_i k(x_i, x) + b$$

Figure 3-15. SVM Classifier Implementation on FPGA.

The DWT coefficients level 3 that are extracted during runtime from received modulated signal are applied to RBF module. RBF module performs the kernel function transformation into separable space. RBF module is constructed with the parameters obtained from training stage of SVM classifier and stored into FPGA design. The output results from RBF module are multiplied by SVM Alphas (Lagrange multiplier) and the results are added to SVM bias (SVM Alphas and bias are stored into SVM on FPGA design). The classification result is compared to a threshold, which is set to zero in this research, so if the output result is positive number, the SVM classifier decision will be

BPSK class (-1), otherwise, the SVM classifier decision will be QPSK class (1). The whole architecture of SVM classifier is described and illustrated as seen in Fig. 3-16.

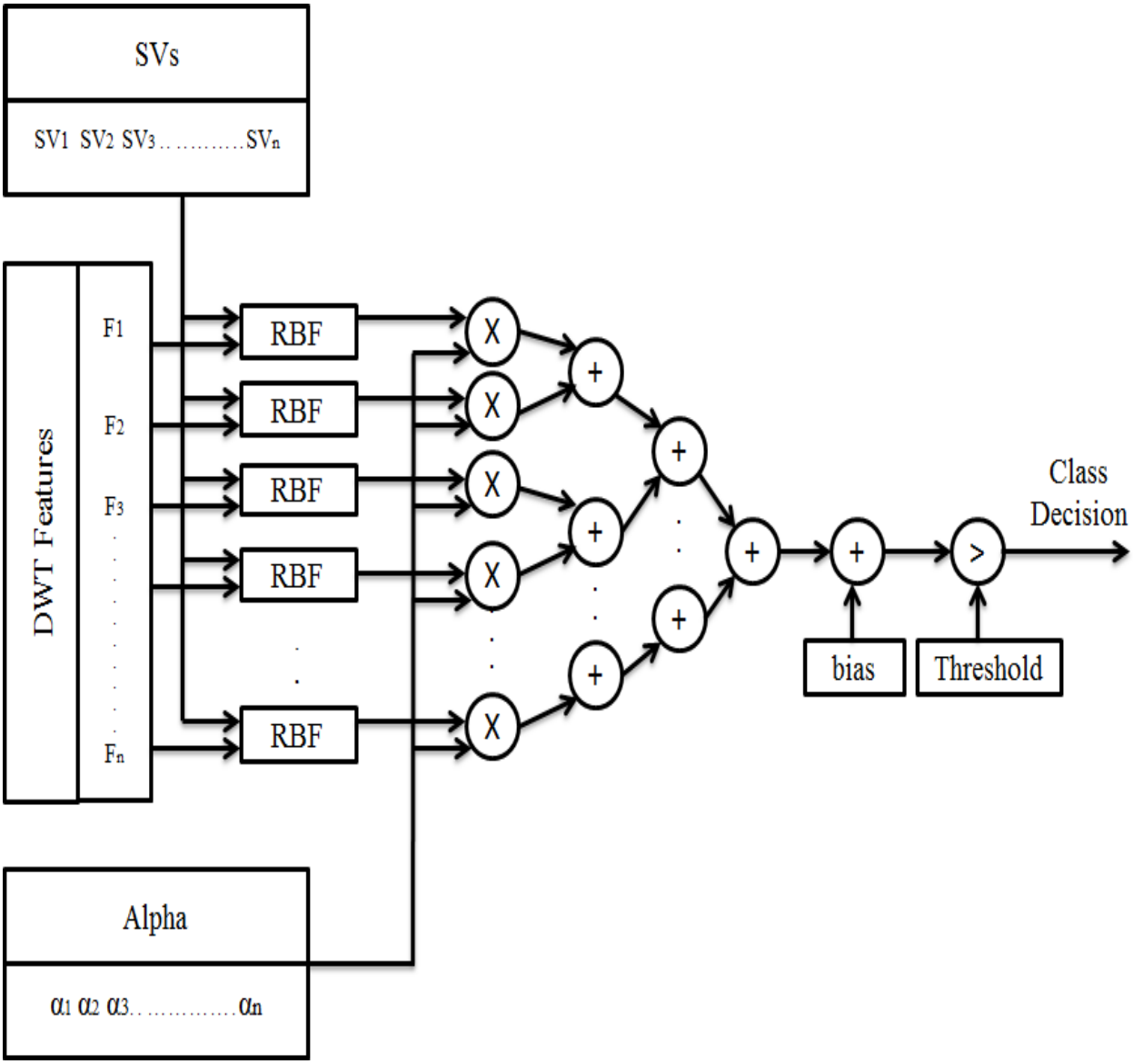


Figure 3-16. SVM Architecture.

3.3. Evaluation System Design

To evaluate the performance of DPRDS, it is required to implement a system that generates BPSK and QPSK modulated signal to simulate the presence of real received modulated signal. Implementation of evaluation system allows the calculation of recognition rate, classification time, and reconfiguration time.

The design of evaluation system (MPSK modulator) consists of Linear Feedback Shift Register (LFSR) to generate sequential input data bits which acts as an input to BPSK phase selector pin of multiplexer, and as an input to serial to parallel converter. the serial to parallel converter converts input serial bits to di-bits that acts as input to QPSK phase selector pins of a multiplexer and according to the value of these bits a sinewave of different phase will be generated. Numerical Controlled Oscillator (NCO) is used to generate four sinewaves of different phases 0º, 90º, 180º, and 270 º as seen in Fig. 3-17.

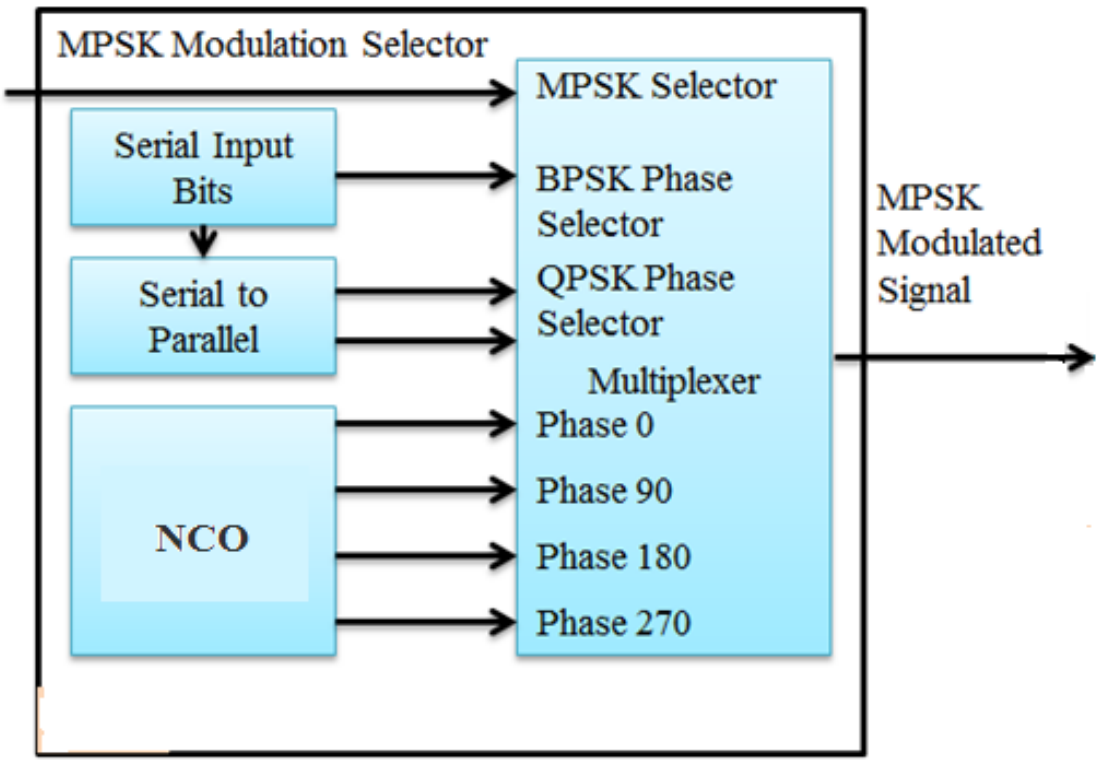


Figure 3-17. Evaluation System Design.

The selection of the required modulation scheme type is occurred by using MPSK modulation selector which is a user switch. According to that selection, the multiplexer receives the input bits from BPSK phase selector pin or QPSK phase selector pins. According to the value of these input bits, a sinewave of specific phase shift is generated each clock cycle of the input bits and so on till the generation of the corresponding BPSK or QPSK modulation schemes.

For example, if the value of MPSK modulation selector is 1, so the targeted output is QPSK modulation scheme. And the QPSK phase selector value is changed according to input bits value every two clock cycles of input bits. Then according to each di-bit value one sinewave of specific phase shift is generated to construct QPSK modulation scheme as shown in Table 3-3, and 3-4. The BPSK and QPSK modulated signal output is illustrated in Fig. 3-18, and 3-19.

Table 3-3. MPSK Modulation Selector Value and Corresponding MPSK Modulation Type

MPSK Modulation Selector	MPSK Modulation Type
0	BPSK
1	QPSK

Table 3-4. MPSK Modulation Selector Value and Corresponding MPSK modulation type

BPSK Selector Value	BPSK Phase	QPSK Selector Value	QPSK Phase
0	0	00	0
		01	90
1	180	10	180
		11	270

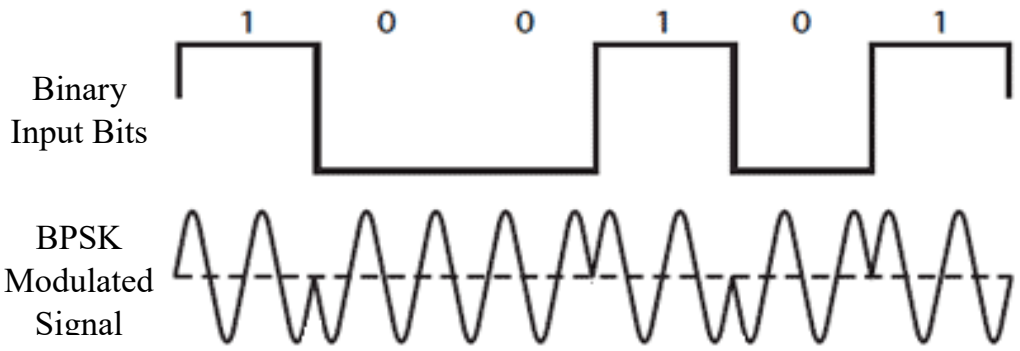


Figure 3-18. BPSK Modulated Signal.

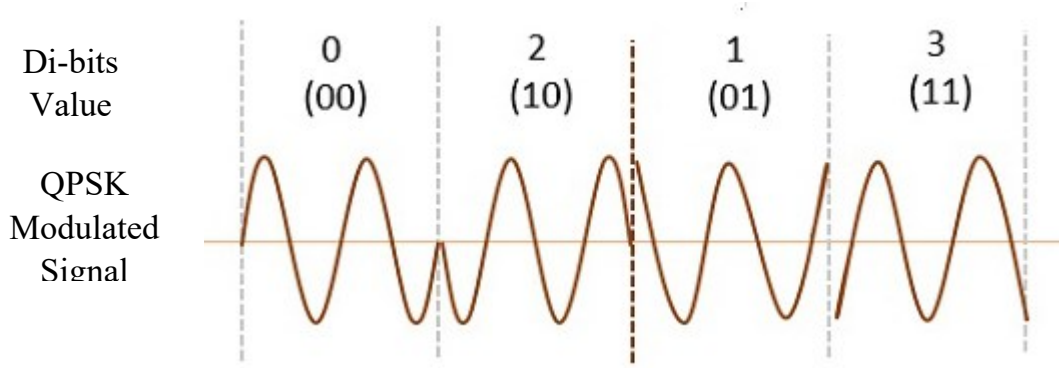


Figure 3-19. QPSK Modulated Signal.

The generation of four sinewaves of different phase is based on the NCO principle. The principle of NCO is to store the sinewave samples in a LUT. The number of samples that represents a sinewave is determined by the difference in phase between two adjacent samples and vice versa. For example, if the phase difference between adjacent samples is 45^{o} so, 9 samples are representing a complete sinewave, as seen in Fig. 3-20.

A phase accumulator, tuning word or phase step, and phase offset are used to generate a sinewave starting from a specific sample and phase. Phase accumulator is N-bit register and the number of samples of a sinewave that will be stored in LUT is determining according to accumulator size (2^{N}). So, if phase accumulator is 8-bits size, so the maximum number of samples is 256 samples. The phase step selects which samples stored in LUT will be output and which will be skipped. The output from phase

accumulator acts as an address to the samples in sinewave LUT. Phase offset determines from which address the sinewave samples will be produced as seen in Fig. 3-21 [54].

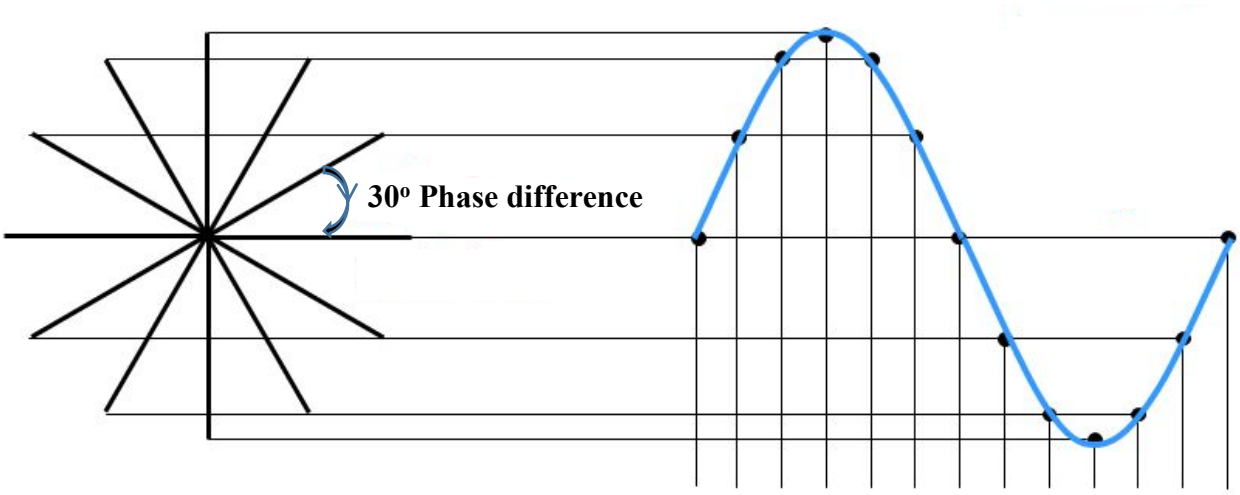


Figure 3-20. Storing Sinewave Samples in LUT.

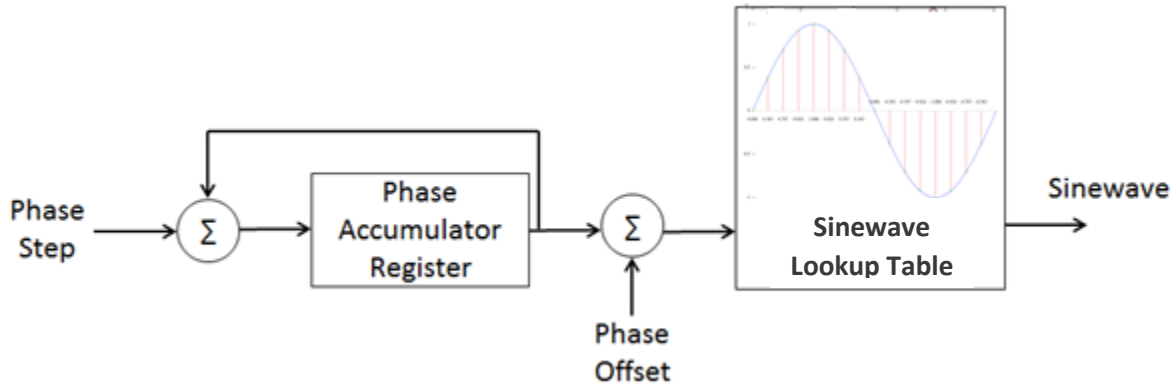


Figure 3-21. NCO Configuration.

The samples of the sinewave are computed using MATLAB tool then these samples are stored into LUT in evaluation system design on FPGA. The value of the required output frequency (f_{out}) determines NCO parameters; phase accumulator size, the step between adjacent samples of the sinewave samples in LUT, and operating clock frequency. In this research, the output frequency (f_{out}) is selected to be 6.25 MHz and the NCO parameters is determined as follows, where 16 samples are used to represent a sinewave for the

simplicity and 4-bit counter is used. The phase step constant is selected to be 2. The clock frequency is 50MHz. The output frequency (f_{out}) is computed according to Eq. (3-12).

$$f_{out} = \frac{Phase\ Step * Clock\ Frequency}{Accumulator\ Size} \quad (3\text{-}12)$$

$$f_{out} = \frac{2*(50*10^6)}{16} = 6.25\ MHz$$

3.4. Performance Evaluation Aspects of DPRDS-Classification System

The evaluation of DPRDS-Classification part is performed by using the BPSK and QPSK modulated signals generated from MPSK modulator. The performance evaluation of DPRDS-Classification part has to be realized with the presence of AWGN with different SNR values to test the accuracy and recognition rates of the system. The summarize of performance evaluation outlines are listed as follows;

- Implementation of evaluation system that generates BPSK and QPSK modulated signals according to outside control.
- Determination of testing configuration of the system and how this system will be tested.
- Calculation of recognition rate of the classifier in the presence of AWGN with different values of SNR using MATLAB.
- Comparing recognition rate with the computation of the same and different techniques in previous studies
- Calculation of the classification time required to classify the modulation scheme of a modulated signal.
- Comparing the resulted BERs of BPSK and QPSK modulations with the theoretical one.
- Calculation of reconfiguration time required to reconfigure reconfigurable partition with corresponding reconfigurable demodulator modules (described in detail in Chapter 4).

- Calculation of FPGA Power consumption and resources utilization of the realized system (described in detail in Chapter 4).
- Comparing resulting demodulated serial data with the original data (described in detail in Chapter 4).

3.4.1. Testing Configuration

The testing configuration block diagram of DPRDS-classification is implemented as shown in Fig. 3-22.

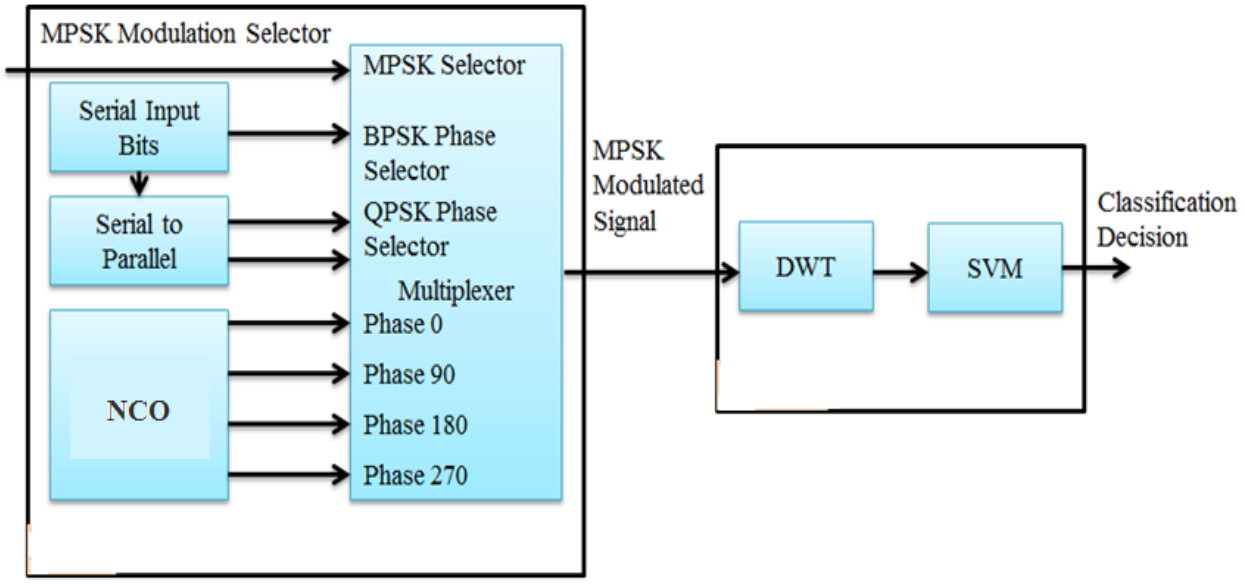


Figure 3-22. Testing Configuration Block Diagram.

From Fig.3-22, it is required to test the DPRDS-Classification part using two FPGA devices; one configured by the evaluation system and the other configured by DPRDS-Classification part. In this dissertation, two Kintex-7 kc7k160tfbg484-1 FPGA devices are used to implement the performance evaluation test.

One Kintex-7 FPGA is used to generate BPSK and QPSK modulated signal according to the MPSK modulation selector (user switch), synchronous clock, and reset signals during testing process. The other Kintex-7 FPGA is used to extract features from the modulated signal, classify modulation scheme, and generate a classification decision that will be

used as the input to DPRDS-DPR part to perform the reconfiguration of the reconfigurable partition by the proper reconfigurable demodulator modules.

The realization of testing system of DPRDS-Classification part during real-time is illustrated in Fig. 3-23 and Fig. 3-24.

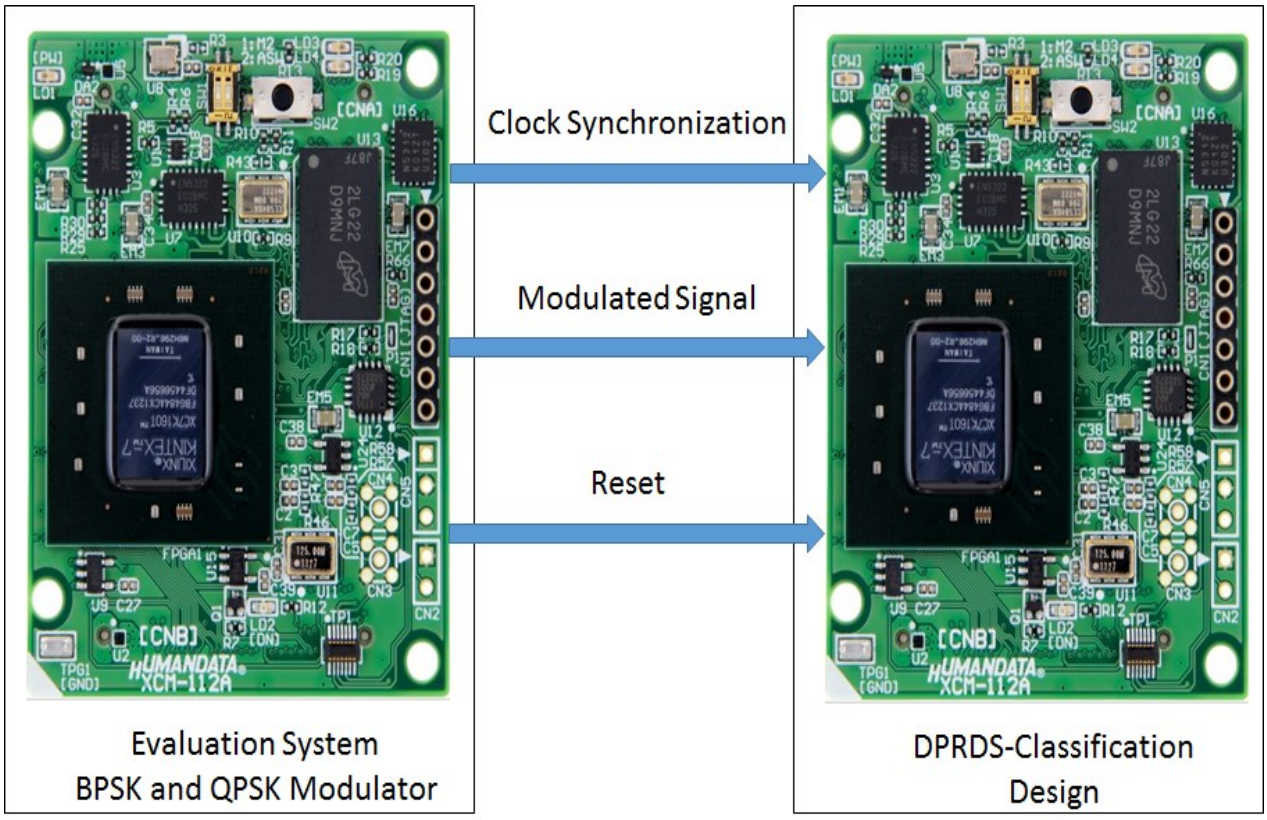


Figure 3-23. Classification Testing Configuration.

The synchronization in telecommunication system is important to prevent the improper recovery of the transmitted bits at the receiver end. A synchronous clock is generated from evaluation system to classification system to achieve this purpose. The system, does not use a Costas loop or PLL for carrier recovery for simplicity.

The BPSK or/and QPSK modulated signal is generated and transmitted to classification system through cables from one FPGA board to another, and the same situation with reset and clock synchronization signals are also transmitted.

After receiving the modulated signal, features have to be extracted and pass through SVM classifier processes till the generation of the classification decision. Two LEDs are used to indicate the classification decision result. If the classification result is BPSK the output will be 01, and if QPSK, the output will be 10, which clearly appear on LEDs. The transmitted and received signal, features extraction and classification processes are monitored using Integrated Logic Analyzer (ILA) IP core on FPGA.

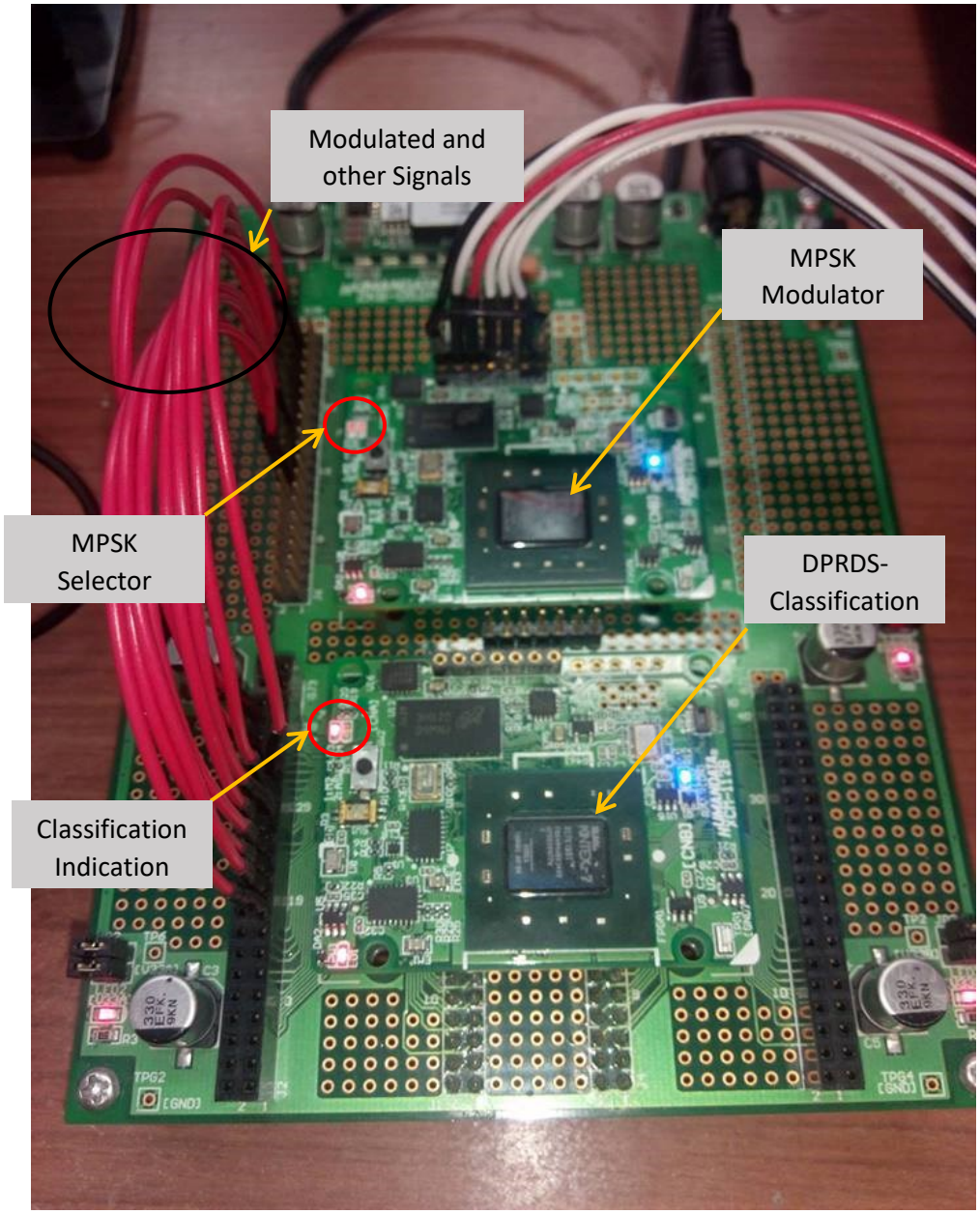


Figure 3-24. Classification Testing Configuration.

3.4.2. Results and Discussion

In this section we describe the simulation and real output results during runtime testing of DPRDS-Classification part on Kintex-7 FPGA using FPGA ILA logic analyzer. The results of MPSK modulator, DWT, SVM classifier, recognition rate, and classification time are described in this section.

The output of MPSK modulator is determined by a user switch on Kintex-7 FPGA board either BPSK or QPSK modulation scheme as seen in Fig. 3-25. The generated signals from NCO and the corresponding generated BPSK and QPSK modulated signals and the recovered original serial input data are resulted in ILA as seen in Fig. 3-26 and 3-27.

MPSK Switch	MPSK Type
0	BPSK
1	QPSK

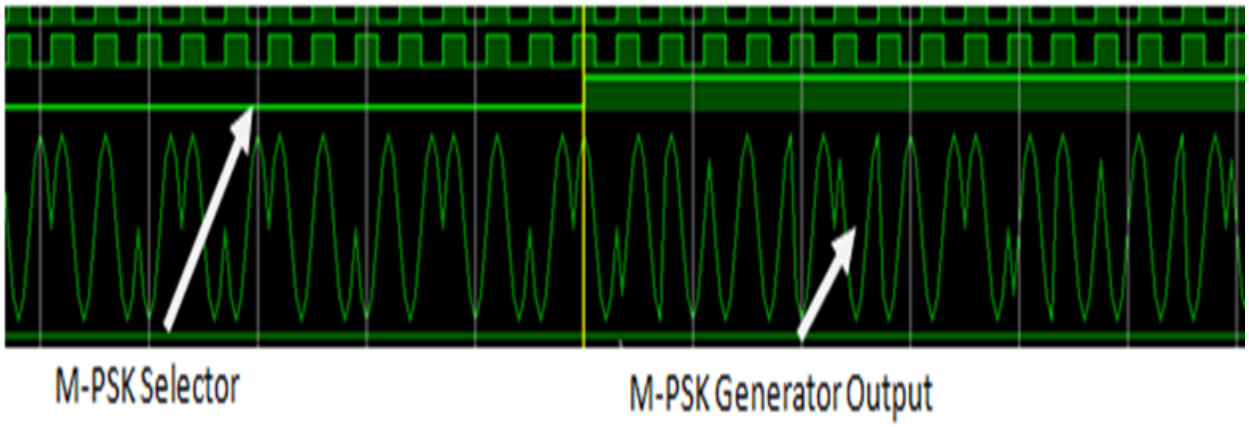


Figure 3-25. MPSK User Switch Value and The Corresponding Modulated Signal.

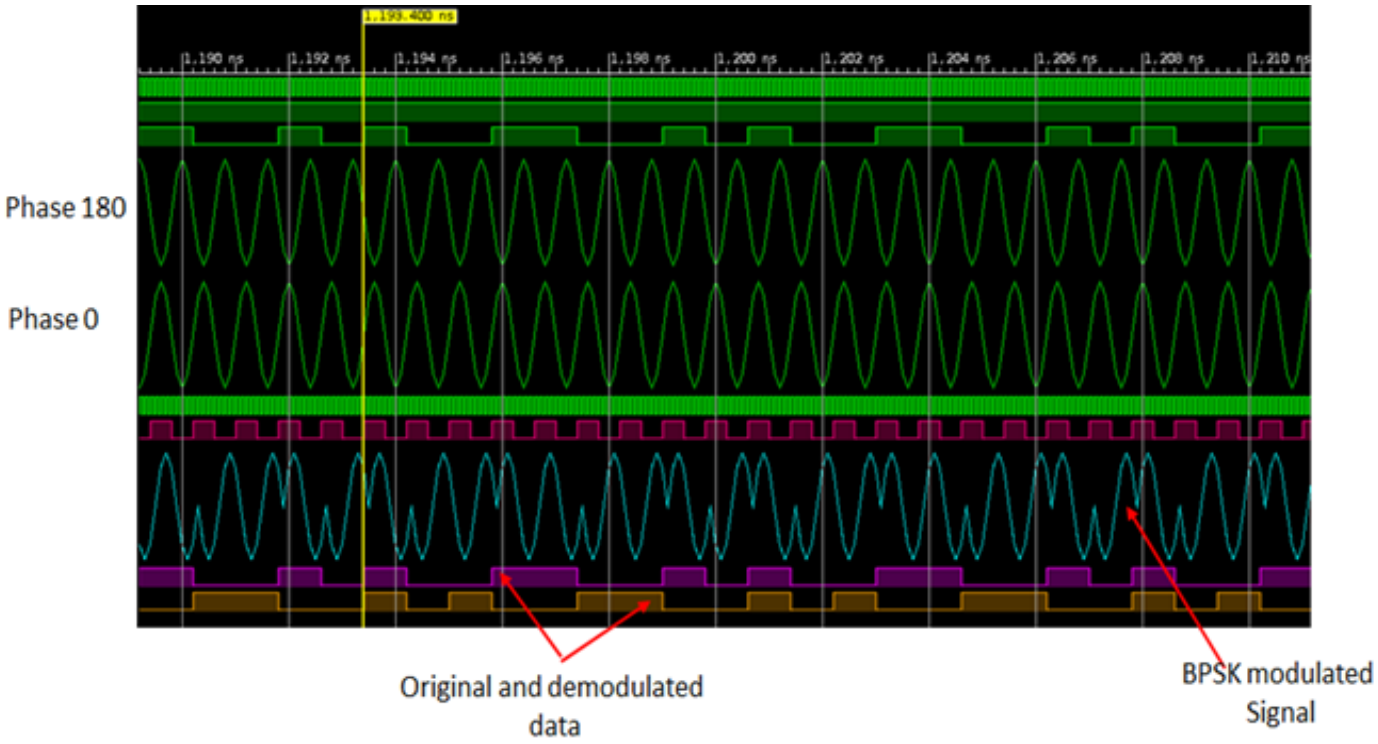


Figure 3-26. BPSK Modulated and Demodulated Signal.

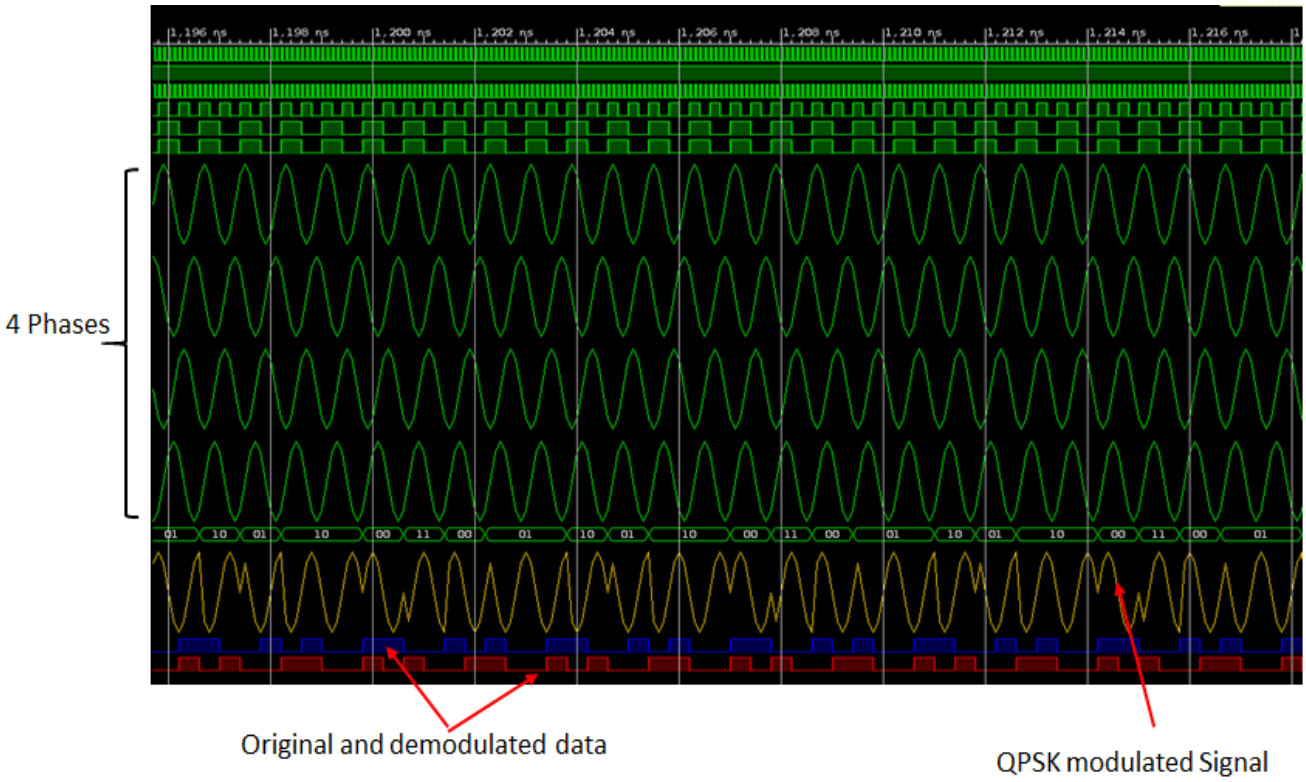


Figure 3-27. QPSK Modulated and Demodulated Signal.

The Bit Error Rate (BER) of the demodulated data measures the performance of the both BPSK and QPSK demodulators. MATLAB “bertool” tool is used in a Simulink model to plot theoretical BER for both BPSK and QPSK modulation, BER of the implemented BPSK, and BER of the implemented QPSK to compare among all of them.

The BER comparison, as seen in Fig. 3-28, shows that the BERs of the implemented BPSK and QPSK modulations are acceptable regarding to the theoretical BER.

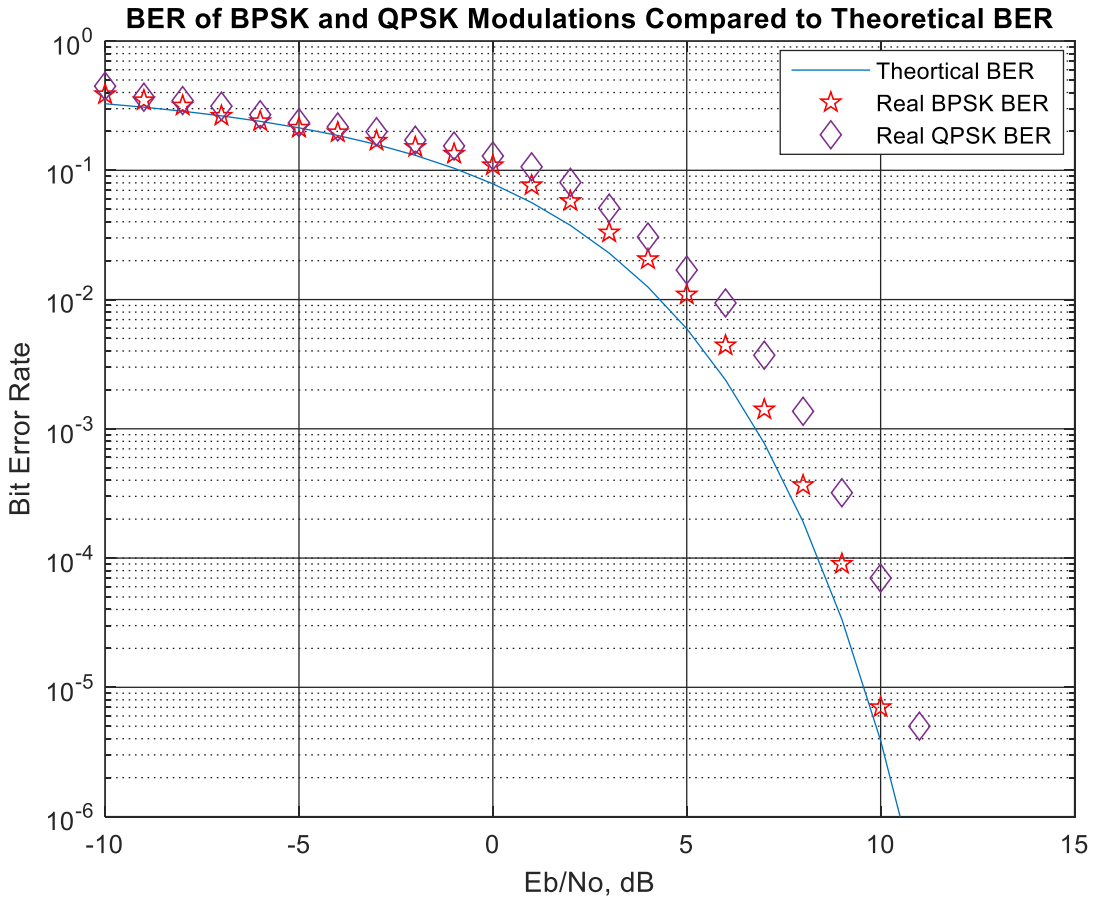


Figure 3-28 BER Comparison between Theoretical and Real BPSK and QPSK.

For DWT implementation, the coefficients of LPF and HPF filters that are used for features extraction process are obtained from MATLAB function “wfilters”, then these filters coefficients are used to set the coefficient parameters during the design and implementation of FIR filters on FPGA to generate DWT coefficients level 3. The computed features for LPF and HPF using MATLAB is listed in Table 3-5 and discrete sequence representation of these coefficients is seen in Fig. 3-29.

Implementing both training and classification stages of SVM classifier on Kintex-7 FPGA increases the utilized resources for the design which increases design complexity and power consumption. Instead, VIVADO 2015.4 and MATLAB r2015b collaboration is used to design DPRDS-classification system to minimize the usage of the utilized resources. The training process has performed by using MATLAB and only the resulting parameters of the constructed SVM (SVs, Alphas, and bias) are used and stored during the design and implementation of the SVM classifier (classification stage) on FPGA.

Table 3-5. LPF and HPF Coeficients

Daubechies (db5)	
LPF Coefficients	HPF Coefficients
0.0033	-0.1601
-0.0125	0.6038
-0.0062	-0.7243
0.0775	0.1384
-0.0322	0.2422
-0.2422	-0.0322
0.1384	-0.0775
0.7243	-0.0062
0.6038	0.0125
0.1601	0.0033

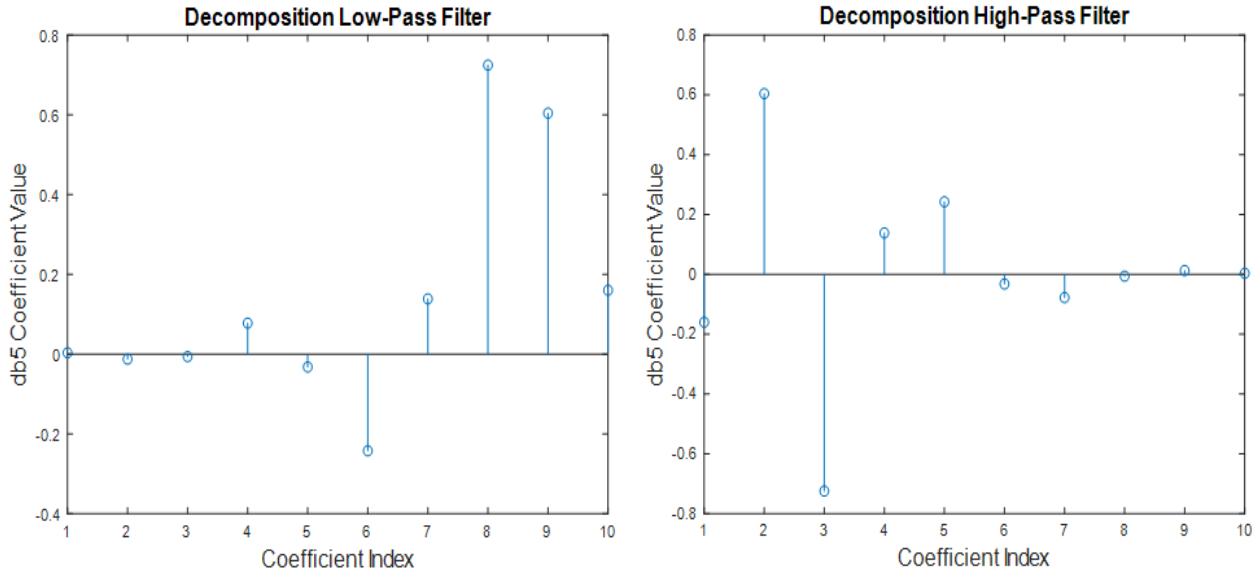


Figure 3-29. Discrete Representation of LHP and HPF Coefficients.

Table 3-6 shows a time series of the extracted features of both BPSK and QPSK modulations as output from the DWT third stage.

Four SVM kernel functions are tested by using MATALB to select the proper one by applying many trails until find kernel function which has high classification accuracy. To perform this test, the extracted features (DWT coefficients level 3) from both BPSK and QPSK modulated signal during runtime (on FPGA) are stored into text file. Then MATLAB loads these features file and create a matrix for these features.

This matrix contains two columns; one contains the extracted features and the other contains the corresponding class label for each modulation type (BPSK or QPSK). i.e., features that are extracted from BPSK assigned to -1 class and that from QPSK assigned to +1 class. Then these features are dividing into two groups; one for train SVM and the other for the testing, using “svmtrain” and “svmclassify” functions. Both “classperf” and “correctrate” functions are used to obtain the accuracy of each kernel function. The output parameters (SVs, Alphas, bias), which are stored into SVM database, are used

during the implementation of the SVM classifier on FPGA. An example of the training process of the SVM is shown in Fig. 3-30.

Table 3-6 BPSK and QPSK Extracted Features form DWT Level 3.

DWT Coefficients Level3		
No.	BPSK	QPSK
1	0.30345	0.30350
2	0.31568	0.31570
3	1.87145	1.87104
4	-1.15693	-1.21985
5	2.35978	1.55737
6	-0.41676	-1.78774
7	-1.88661	1.97410
8	0.30214	0.54455
9	1.90090	-1.35940
10	-0.26326	1.77623
11	-0.88683	-0.92732
12	2.31185	1.44309
- -	- -	- -
21	0.14785	2.22777
22	-1.04387	1.37669
23	1.35286	0.10046
24	-1.41577	-1.36278
25	2.44605	0.87444

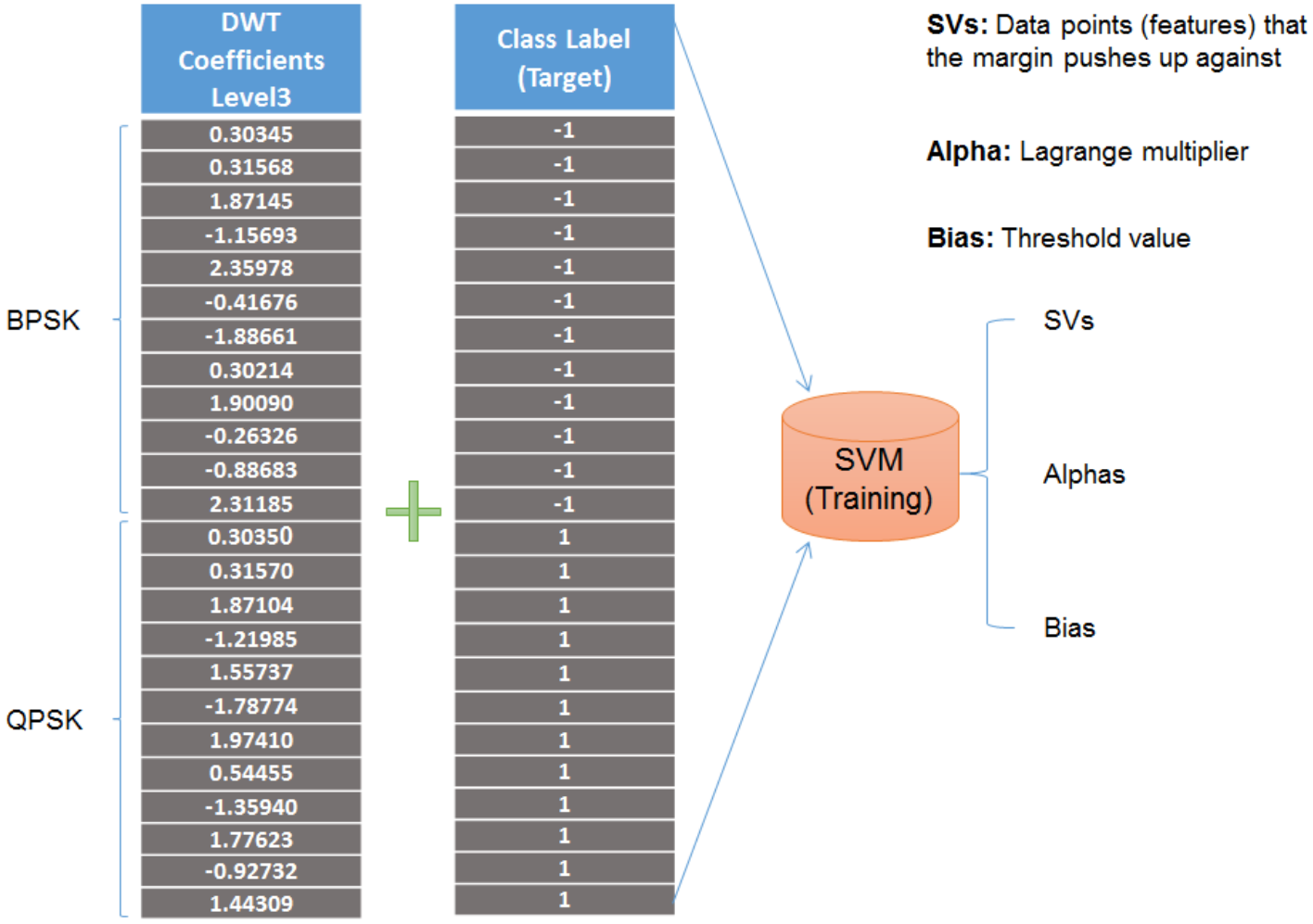


Figure 3-30 Example on the Training Process of the SVM

The resulting accuracy from this test is obtained after 23 trails using 25 features for each modulation scheme as seen in Fig. 3-31.

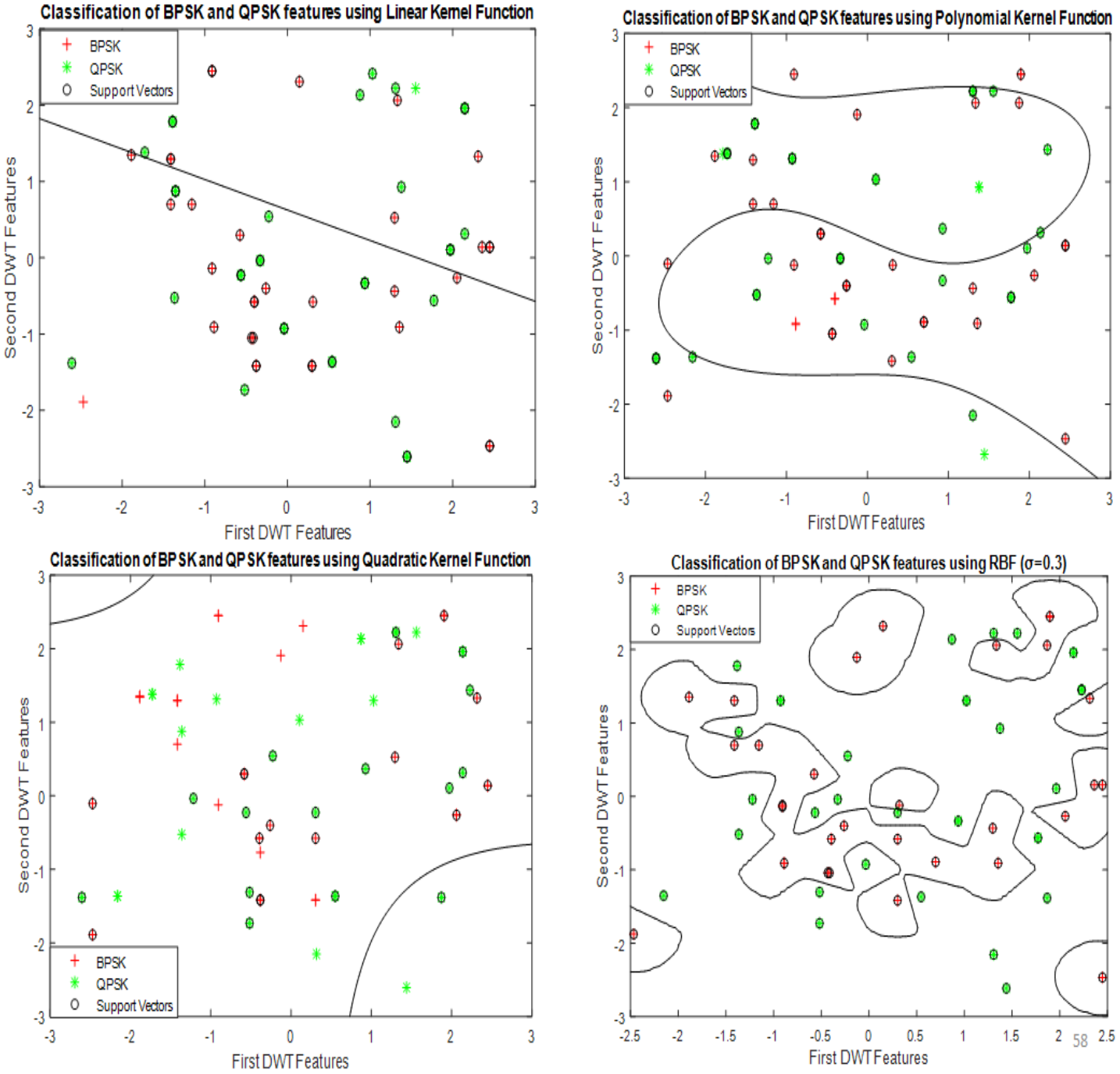


Figure 3-31. Comparison among the Accuracy of Different Kernel Functions.

The results of classification accuracy of the kernel functions show that RBF kernel function has the highest classification accuracy. For RBF, different values of σ parameter are tested until getting the σ that achieves the highest accuracy which is realized for σ = 0.3 as in Table 3-7. The reason that RBF kernel function has high classification accuracy than the other kernel functions is that RBF uses radial (nonlinear) separator to solve the optimization problem by make proper boundaries between the two classes (green circles of QPSK features and red circles of BPSK features) which do not exist for the others kernel functions. The inconsistent results fail to classify between BPSK and QPSK features because they use unsuitable separators for the existing problem as seen in Fig. 3-31.

Table 3-7. RBF Classification Accuracy for Different σ Values

RBF Kernel Function	
Sigma Parameter (σ) Value	Classification Accuracy
0.1	80.96
0.2	87.30
0.3	94.44
0.4	81.30
0.5	77.19

Many trails have been performed to select the proper value of σ. σ value can be selected from 0 to 1 range. It can be more than 1 but no success occurs to classify among different classes. In this dissertation, several values of σ are tested to get the proper value of σ.

A comparison among different values of σ is performed as shown in Fig. 3-32. The best value of σ that has the highest classification percent and accuracy (94.44) is 0.3. If σ decreases or increases around 0.3, the accuracy decreases.

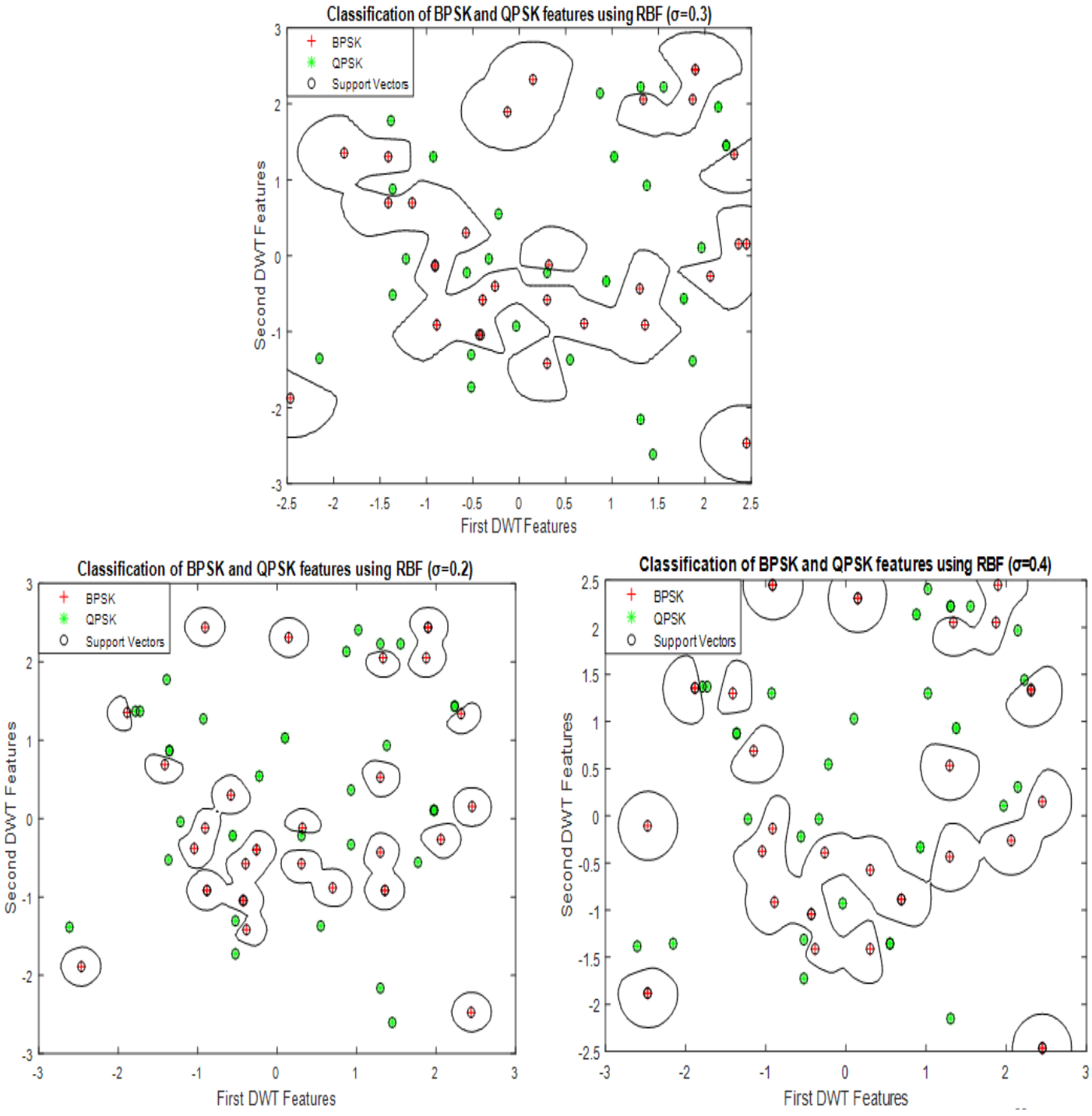


Figure 3-32. Comparison among the Accuracy of Different Values of σ.

The classification time measures the speed of DPRDS-Classification part and evaluate the suitability of this system for the application purpose. this shows the elapsed time starting from the selection of the modulation scheme passing through features extraction and the classification calculations processes till the generation of a classification decision by SVM classifier is as seen in Fig. 3-33. The classification time of DPRDS is 6.7 µsec as seen in Fig. 3-33.

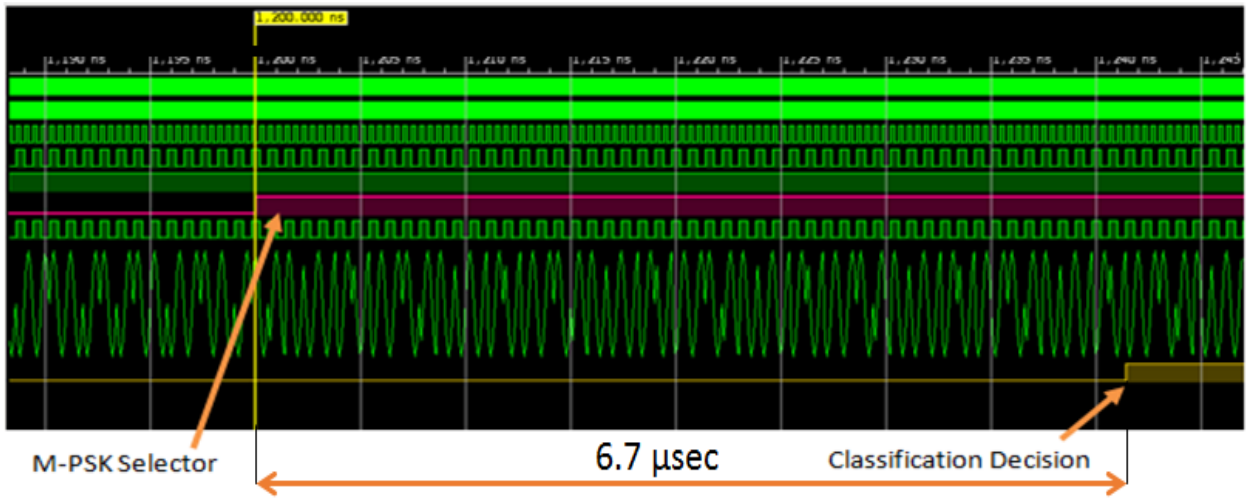


Figure 3-33. DPRDS Classification Time.

The performance of SVM classifier is realized by applying Additive White Gaussian Noise (AWGN) of SNR from -10 to 20 dB on the received modulated signal to simulate the receiver noise. MATLAB is used to simulate AWGN that can affect received modulated signal at receiver side. 1,000 samples of both BPSK and QPSK modulations are used to test the performance of SVM classifier. 100 trials have been applied on the resulted features of these samples for each modulation technique (at each SNR) to determine the recognition rate of both modulations in the presence of AWGN with SNR from -10 to 20 dB.

First, the received modulated signal is loaded into MATLAB by using blackboxs in Simulink to load the corresponding VHDL modules to generate the same modulated signal generated using FPGA. After the SNR is added to the modulated signal, the output

is passed to SVM classifier, and then the output of SVM classifier is passed to a module calculated the recognition rate. the number of correct classification trails expresses how successfully SVM classifier can classify the modulation type of the received signal. The result of the recognition rate of both BPSK and QPSK modulation schemes is as shown in Fig. 3-34.

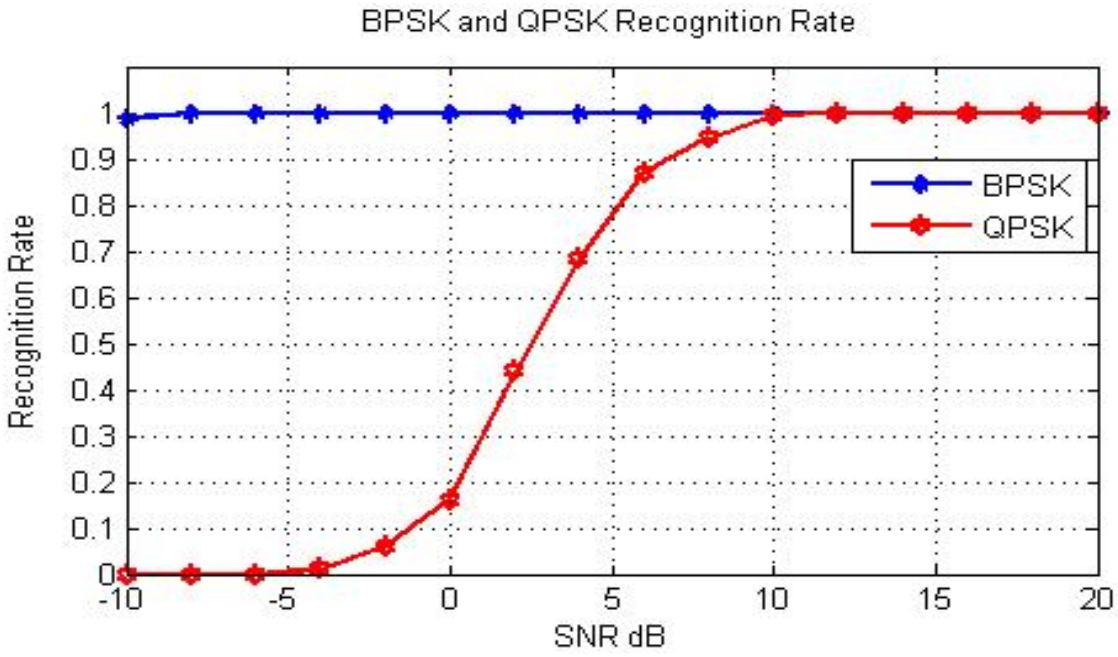


Figure 3-34. SVM Classifier Recognition Rate for BPSK and QPSK.

Fig. 3-34 shows that at low SNR the SVM classifier classifies BPSK modulation scheme with 98.7% recognition rate then reach 100% at 0 dB, in contrast, the SVM classifier fails to recognize QPSK modulation scheme at low SNR but the recognition rate increases gradually till reach 100% at 10 dB. The calculation of the recognition rate is based on number of correct recognition trials to the total number of trials as expressed in Eq. (3-13).

$$Recognition\ Rate(RR) = \frac{Number\ of\ Correct\ Classification\ Trials}{Total\ Number\ of\ Classification\ Trials} \quad (3\text{-}13)$$

For more evaluation of SVM classifier of DPRDS system, a comparison between the results obtained in this dissertation and the results of previous studies is done.

A comparison between the classification results of each modulation scheme and the classification results of 4 previous studies is performed to show how the SVM classifier can be applicable for the classification of modulation types. In [56], a Kernel-based Generalized Discriminant Analysis (KGDA) system for the classification of the several modulation types (as BPSK and QPSK) was proposed where statistical and the spectral features of different modulation schemes were used to train the SVM classifier. Two sample groups were used to test the recognition rate of the SVM classifier in the presence of AWGN, 1,024 and 4,096 samples. The results showed as the number of the samples used to train the SVM increased the recognition rate increased. In [57], 6 spectral and 12 statistical features were used to discriminate among several digital and analog modulation types, and a SVM classifier was used to classify them in the presence of AWGN. The results showed that the classifier had a good performance with 85% to 98% classification probability. In [58] a combination of the fourth, the sixth and the eighth order of moments and cumulants of the modulated signal were used as an input features of a multiclass SVM to differentiate among several modulation types. 6,000 samples were used to test the recognition rate of the classifier in the presence of AWGN. The results showed that the classifier had good classification performance starting from 0 dB.

In [59], the frequency-smoothing method was used to extract four spectral coherence characteristic parameters to represent 6 modulation types. 1,024 samples were used to test the performance of the classifier in the presence of both AWGN and multipath fading. The results showed that the total success rate was above 92.8% at SNR=4 dB and increased gradually as SNR increased. In [60], 6 cumulant coefficients values were used as unique features of 9 modulation types, and in the classification stage these features produced distance measurements corresponding to each modulation type. 1,000 trials had done for each modulation type to test the classifier performance at SNR from 0 dB – 20 dB and the results showed good classification rate even at low SNR. In [61], the peaks number in the spectrum after the analysis of second, fourth, and eighth nonlinear transformation could distinguish different modulation types according to nonuniform

compressive samples theory. Based on the numbers of discrete of each order of nonlinear transformation and the rough estimation of both carrier frequency and symbol rate, the modulation type could be detected. The comparisons of SVM classifier recognition rate with that of the other previous studies are as seen in Fig. 3-35, and Fig. 3-36.

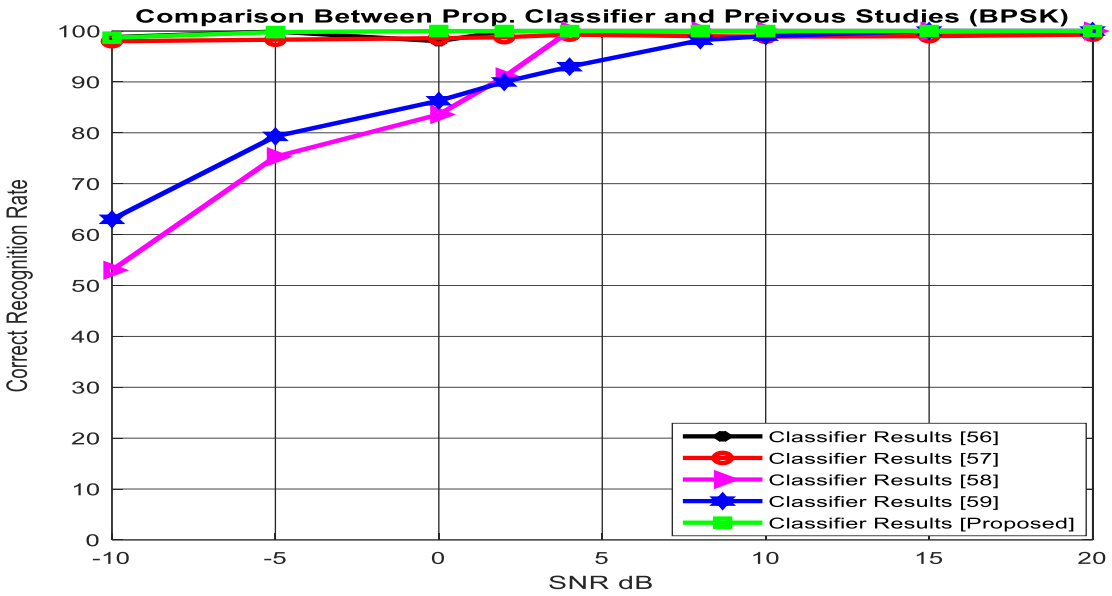


Figure 3-35. Comparison between SVM Classifier and other studies for BPSK

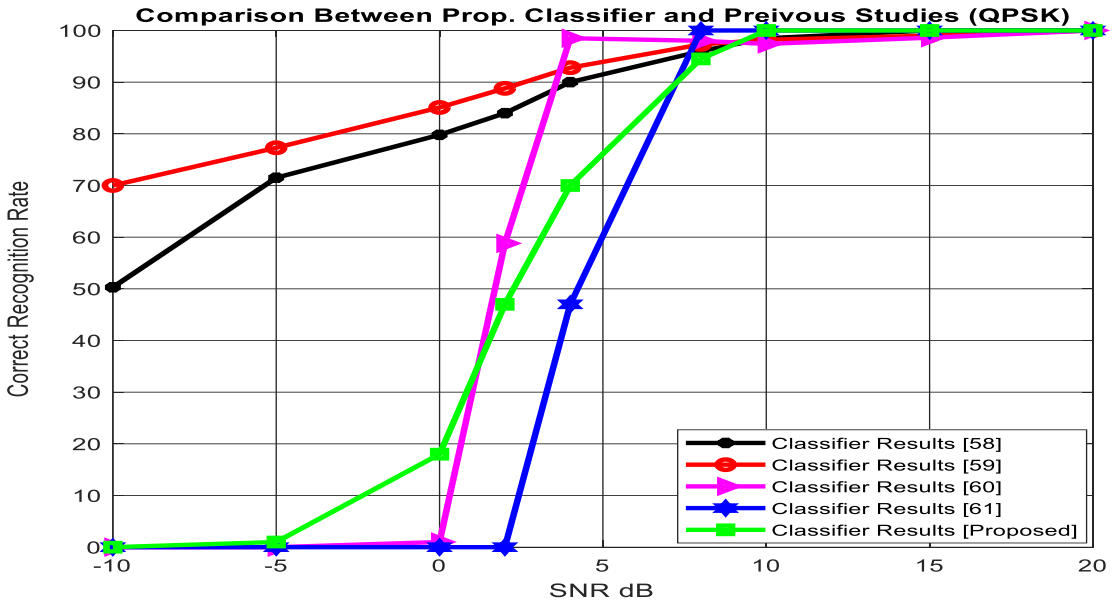


Figure 3-36. Comparison between SVM Classifier and other studies for QPSK

3.4.2.1. Results and Discussion of Testing the DPRDS-Classifier using KIT GCS

The Ground Control Station (GCS) of Kyushu Institute of Technology (KIT) is used to test the performance of the classifier part of the DPRDS design. The GCS of KIT uses to communicate with Horyu-4 and the series of BIRDS satellites. Horyu-4 satellite is a KIT satellite orbits around the earth in Low Earth Orbit (LEO). The purpose of this satellite is to monitor the discharge effects on the solar panels in space. BIRDS project is a multinational program called joint global multi-nation birds satellite project, to help space emerging countries all over the world to build their first satellite as a collaboration between KIT, which supports the design and fabrication of the satellite, and the space emerging countries. GCS communicates with the satellites using UHF/VHF and S-band frequencies. The BPSK and QPSK modulations are commonly used for S-band communications which are the same modulations schemes used in this work. Only the classification test of BPSK modulation is done by using the received BPSK modulated signal from Horyu-4 satellite at the GCS of KIT because Horyu-4 satellite uses only the BPSK modulation scheme for downlink of the payload data to the GCS.

The 2.4 GHz modulated signal is received via S-band parabolic antenna to be passes through optical to RF converter, and then the output acts as an input to ICOM R-9500 receiver which down converts the frequency to 10.7 MHz. The 10.7 MHz signal is then applied to ADC expansion module to convert it from analog to digital format to be suitable for the processing on FPGA. The output signal from ADC module is connected to Kintex-7 FGPA board where the classification part of DPRDS design is implemented. The FPGA is connected to a laptop running VIVADO 2015.4 tool to read the results during run-time using Integrated Logic Analyzer (ILA) tool which is implemented into the design. The test configuration is as shown in Fig. 3-37 and Fig. 3-38.

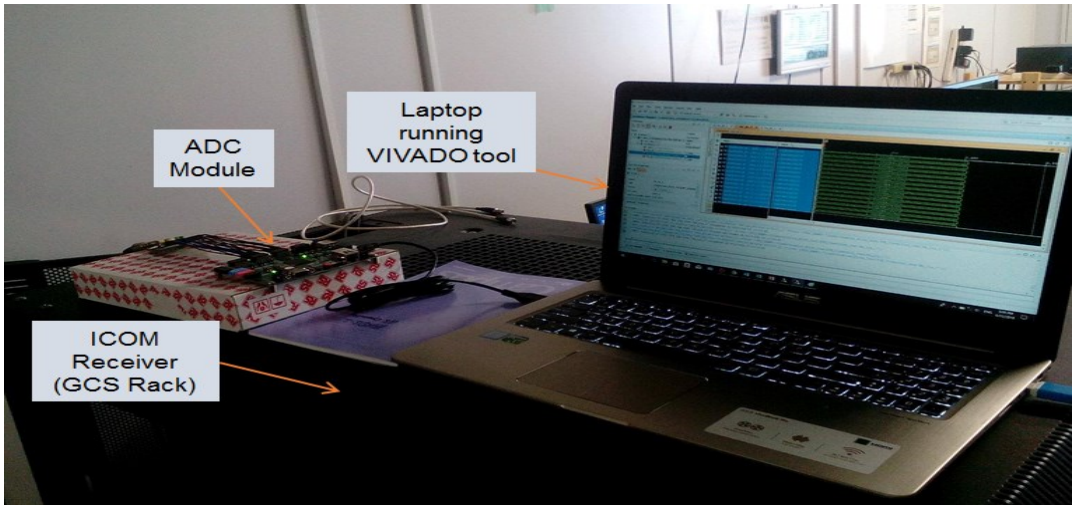


Figure 3-37. BPSK Classification Test Setup at GCS of KIT

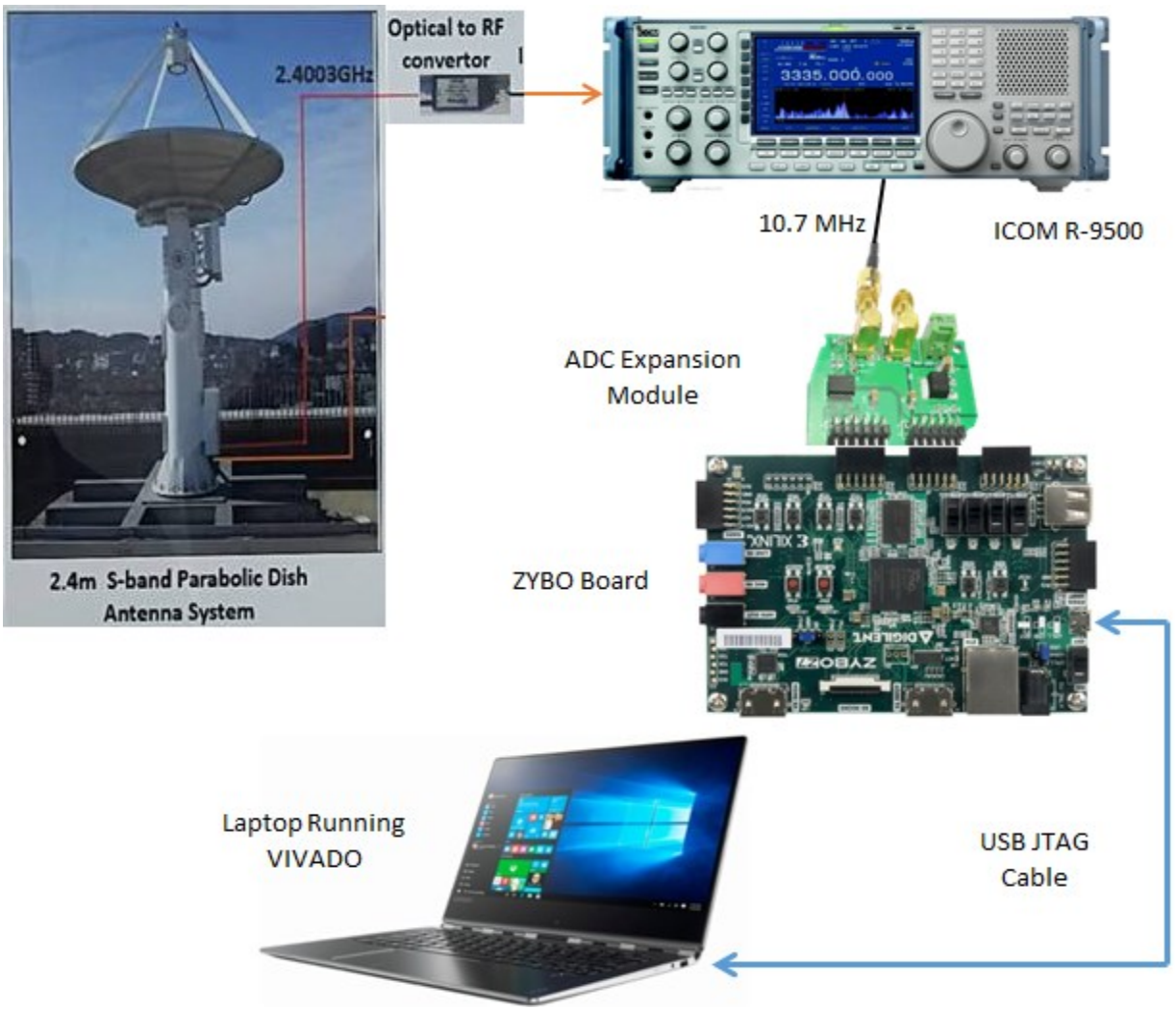


Figure 3-38. The BPSK Classification Test Configuration

During the communication session with the satellite, the output from ADC module will be fed with the input of the features extraction to extract DWT coefficients level3, which are propagated to SVM classifier to decide whether the received modulation is BPSK or not. The extracted features are seen by using VIVADO ILA tool. The values of the extracted features were almost the same as that the classifier was trained on, therefore, the classifier could successfully classify the BPSK modulation. The values of the extracted features can be seen in Fig. 3-39.

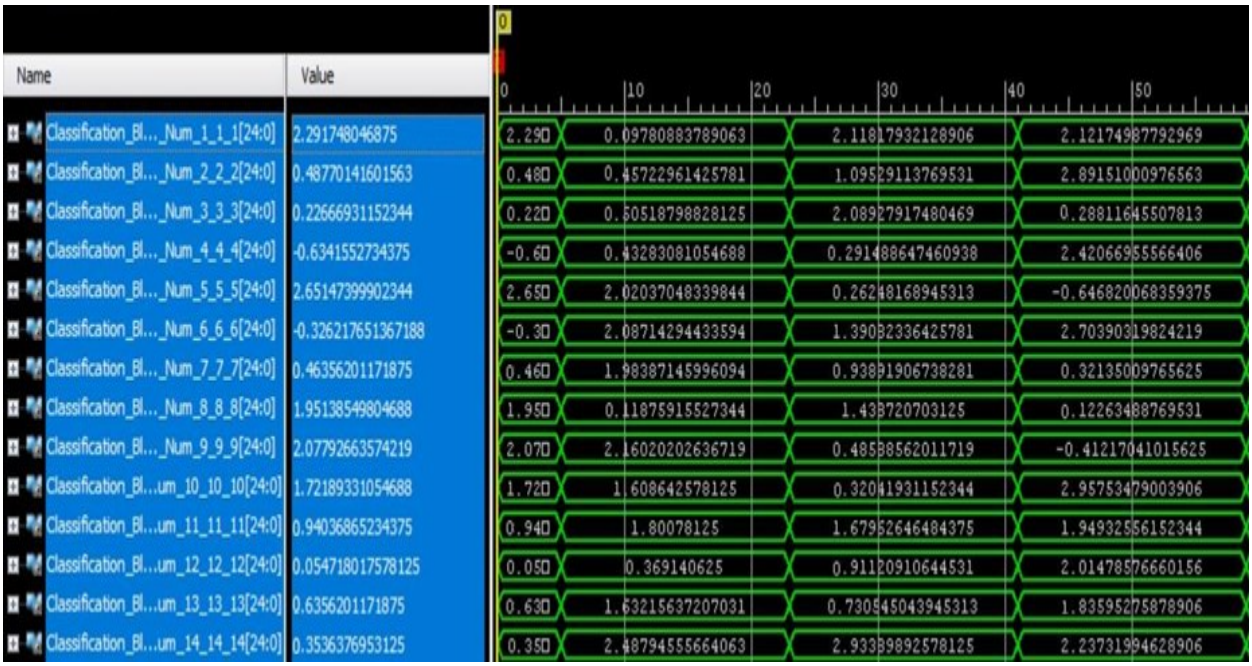


Figure 3-39. The Extracted Features of BPSK Modulated Signal

The output result of the classification process is connected to two LEDs on the FPGA board and the classification result shows that the classifier successes to classify the BPSK modulation of the received signal most of the times but for a few times the classifier failed to recognize the BPSK modulation. One of the reasons is the classifier did not be trained well on the BPSK modulation in the presence of all types of noise like fading and also the Doppler shift effect. So, to implement more accurate classifier it is recommended

to train the classifier with the possible noise and variation that can affect the received signal.

The DPRDS-classifier part of the design has been tested by using the received BPSK modulated signal at KIT's GCS which is used to communicate with Horyu-4, and BIRDS projects satellites. Horyu-4 satellite is using the BPSK modulation scheme to downlink the payload data. So, I used Horyu-4 satellite received signal to test the classifier part of the DPRDS design. The reason for that can be described as the classifier did not trained well on the BPSK modulation in the presence of Doppler shift effect, fading problems and the real noise of the channel which make the classification of BPSK modulation a little bit difficult for these few times.

In this chapter, the classification part of DPRDS system is described. Using 3 stages of DWT is used as features extractor where these features are passed to SVM classifier which in turns classifies the modulation technique of the modulated signal. The evaluation of the classification system is performed with the presence of AWGN of SNR -10 to 20 dB. The results show the high performance of the system even at low SNR (98.7% at -10 dB) in case of BPSK modulation type and stating from 0 dB (18% at -10 dB) in case of QPSK.

In the next chapter, the dynamic partial reconfiguration design of DPRDS is described. The DPRDS-DPR design part reconfigures the reconfigurable partition with the corresponding demodulator. Dynamic partial reconfiguration methods, implementation of BPSK and QPSK demodulators, and performance testing are described in next chapter.

Chapter 4 : DPRDS - DPR

In this chapter, the design and implementation of both DPR design part of DPRDS system, and BPSK and QPSK demodulators are described. As mentioned in chapter 3, DPRDS consists of two parts; ADMC and DPR parts. ADMC or classification part of DPRDS is already described in Chapter 3. In addition, the evaluation method of DPR design of DPRDS is described in this chapter to evaluate the performance of the system and to show the real output results after implementing the system on FPGA.

In this chapter we will discuss the design flows of partial reconfiguration process, and reconfiguration modes and ports. Two design flows are used for partial reconfiguration; modular based and difference based partial reconfigurations. In this chapter we will discuss both methods and which one is suitable for the implementation of DPRDS-DPR. Different reconfiguration methods can be used as self-reconfiguration and dynamic partial reconfiguration. For the purpose of this research self-reconfiguration is used.

4.1. DPRDS-DPR

DPR design is segmented into 5 modules; Partial Reconfiguration Controller (PRC) IP core, External Memory Controller (EMC), Buffer, and demodulators modules. PRC IP is responsible of performing and controlling the partial reconfiguration process of the corresponding demodulator and acts as intermediate stage between configuration files stored into external memory and ICAP module. EMC fetches the corresponding reconfigurable module or partial reconfiguration files from external memory and passes these files to PRC according to request from PRC.

ICAP module is used to internally reconfigure a specific location on a FPGA with a part of the implemented design. ICAP reconfigures the dynamic section or reconfigurable partition with the corresponding demodulator during runtime. The buffer is used to store

the received modulated signal during the classification and reconfiguration processes to prevent the loss of any samples of the received modulated signal.

The sequence of the DPR operation starting from the reception of the modulated signal till the regeneration of reconfiguration partition with the corresponding demodulator and the DPR modules and components used to perform the reconfiguration process highlighted in yellow is illustrated in Fig. 4-1.

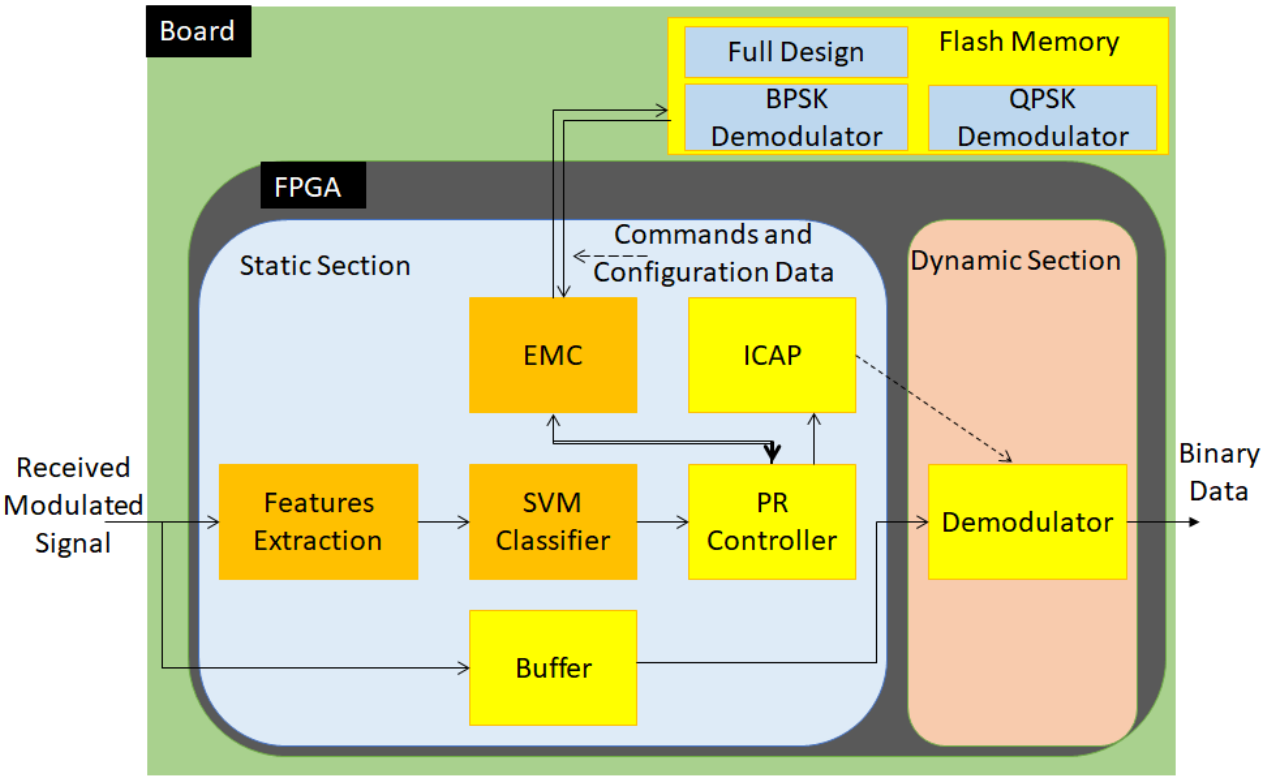


Figure 4-1. DPRDS Design Block Diagram.

The features of the received modulated signal have to be extracted to pass to SVM classifier and generate the classification decision or which modulation scheme (BPSK or QPSK) is received as described before in Chapter 3. Based on this decision, PRC IP which has the addresses of each partial bit-stream file of each demodulator in the external flash memory sends the corresponding memory address of the partial bit-stream to EMC.

EMC in turns fetches the suitable Reconfigurable Module (RM) or partial bit-stream file (demodulator) from external flash memory, and then sends this file to PRC IP.

External flash memory includes full configuration file, and BPSK and QPSK demodulator's partial bit-stream files, where full configuration file contains the whole design of DPRDS system and is loaded at the startup of FPGA.

After receiving the demodulator partial bit-stream file from EMC, PRC IP sends this file to ICAP interface to reconfigure the dynamic section (reconfigurable partition) with the suitable demodulator. The buffer size is selected to be big enough to store all the samples of a received modulated signal during the classification and reconfiguration processes.

4.2. Dynamic Partial Reconfiguration

Xilinx provides partial reconfiguration (PR) feature on FPGA which makes FPGA widely deployable in different applications especially in communication field [42], [45]. This feature allows user to reconfigure FPGA many times and at any time during runtime. PR feature is available in the recent FPGA devices which reconfigures a specific logic area or location called reconfiguration partition (dynamic section) by a reconfigurable module during runtime while the remaining design sections (static sections) is running without any effects on their performance [62].

4.2.1. Dynamic Partial Reconfiguration Methodology

There two methodologies are used to achieve partial reconfiguration; dynamic partial reconfiguration and dynamic partial reconfiguration self-reconfigure as shown in Fig. 4-2. The dynamic partial reconfiguration can be implemented by using external microprocessor to fetches the required partial bit-stream file from external memory through JTAG interface.

In self-reconfiguration of FPGA, an internal microprocessor or IP core controller is used to reconfigure the FPGA via ICAP interface by partial bit-stream file stored into external memory during runtime.

In this dissertation, dynamic partial self-reconfiguration of FPGA methodology is used to realize DPRDS system, and a PRC IP core is used instead of using an internal processor (Xilinx Microblaze).

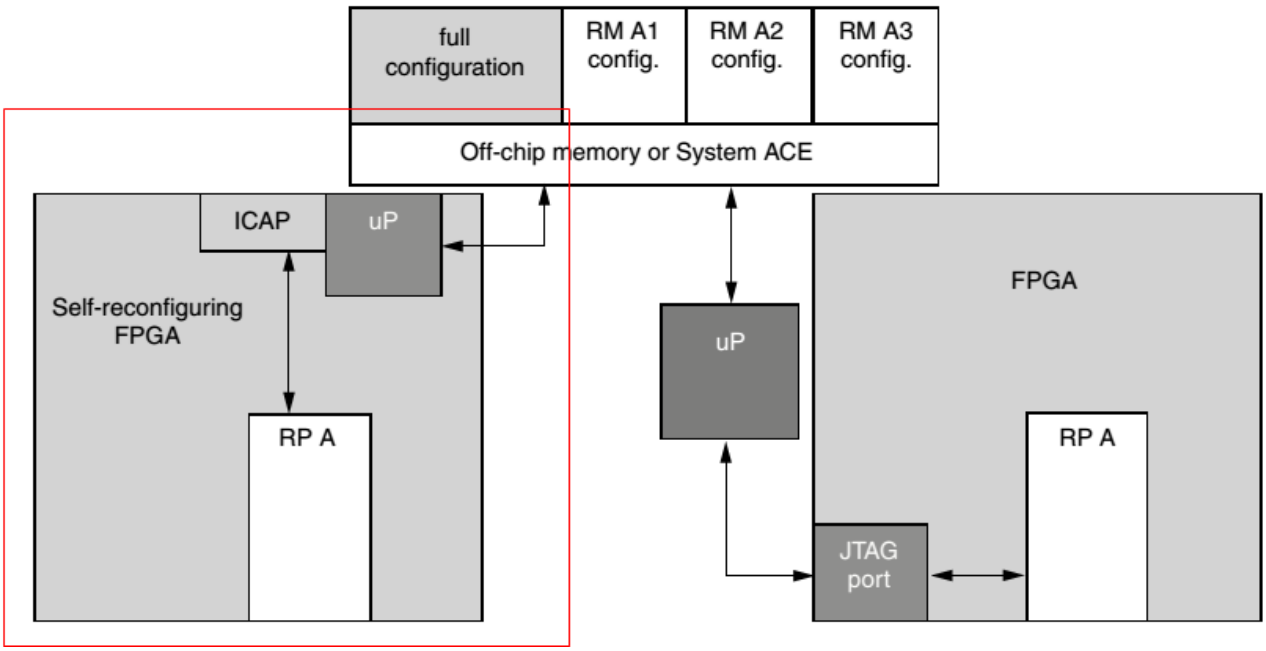


Figure 4-2. Reconfiguration Methodologies.

4.2.2. Dynamic Partial Reconfiguration Ports

Dynamic partial reconfiguration of FPGA can be performed externally or internally. Externally reconfiguration can be implemented by using Xilinx SelectMap port, Serial configuration port, and JTAG (Boundary Scan) port as shown in Fig. 4-2, And internally reconfiguration by using ICAP interface. In this dissertation, ICAP interface is used to reconfigure FPGA internally.

Each of the configuration ports has the maximum clock rate and data width which impact on the maximum bandwidth for each port. As the maximum bandwidth increases as the reconfiguration time decreases. The comparison among different configuration ports is as shown in Table 4-1 [64].

Table 4-1. Comparison Among Different Configuration Ports

Configuration Mode	Max Clock Rate	Data Width	Maximum Bandwidth
ICAP	100 MHz	32 bit	3.2 Gbps
SelectMAP	100 MHz	32 bit	3.2 Gbps
Serial Mode	100 MHz	1 bit	100 Mbps
JTAG	66 MHz	1 bit	66 Mbps

4.2.3. Dynamic Partial Reconfiguration Design Flow

Two types of design flow are used for the dynamic partial reconfiguration process; difference-based PR and module-based PR. Each design flow will be discussed in the following sections.

4.2.3.1. Difference-Based PR

A difference-based bitstream uses to compare the configuration of two design modules. For example, Xilinx BitGen tool compares the bitstream of one module with the native circuit description (NCD) of the other module to generate a bitstream of the differences between these modules. The generated bitstream contains the required modification of the first module to realize the functionality of the other module as seen in Fig. 4-3. The generated bitstream programs only the difference between the two modules. Difference-based PR is only used for designs with two modules only because it requires prior configuration knowledge before generating a differential bitstream [65], [67].

Difference-based PR uses to make small logic changes as changing I/Os, block RAM contents, LUT, flip-flop initialization and reset values, multiplexers, pull-ups or pull-downs on external pins. But it has to be considered that changing of any value that would impact routing is not recommended due to the risk of internal routing.

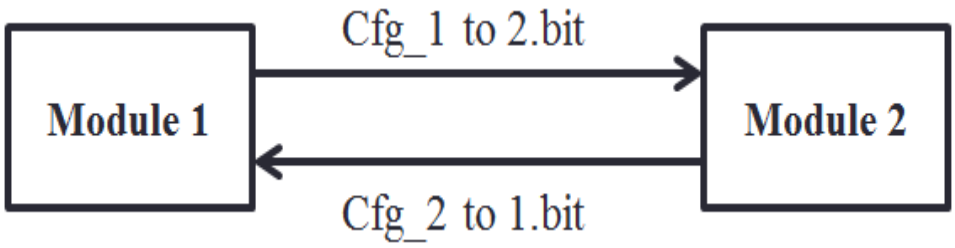


Figure 4-3. Difference Based PR Concept.

The design flow of difference-based PR is to implement static and reconfigurable modules with components constrained at the same location in all the bitstreams to implement the complete bitstreams separately for each RM. Finally the controller computes the difference of two bitstreams to obtain the partial bitstream needed to move from one configuration to the next one as in Fig. 4-4.

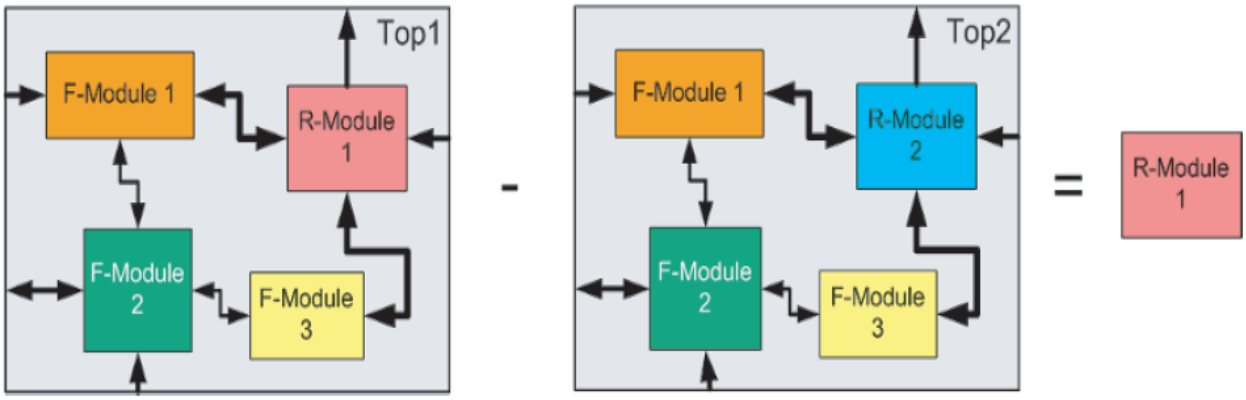


Figure 4-4. Difference Based PR Configuration.

4.2.3.2. Module-Based PR

Module-based partial reconfiguration segments the design into modules. It uses to reconfigure specific modules of the design. The communication between the reconfigurable module and the fixed modules of the design is achieved through bus macros or as called partition pins. It acts as a fixed routing channel that bonds the reconfigurable module with the remaining modules of the design [66].

The insertion of these partition pins into design can be done by whether manually using location constraints offered by the implementation tool or automatically by the tool chain

during implementation of the design. Partition pins are a part of static section of the design and can be placed at any location into reconfigurable module. A partition pin is a LUT in route-through mode [66], [67].

As the RM is allowed to include slices, DSP and block RAM and not allowed to include PLL, I/Os, and MMCM, it means that module-based PR changes only the configuration of these resources included into a RM include. The advantage of module-based PR, it uses for designs with two or more RMs.

The design flow of module-based PR is to divide the design into static and dynamic (reconfigurable) modules to synthesize these modules separately and constraints these modules at a specific location on FPGA. Finally, a complete bitstream is generated (for each RMs) in addition to partial bitstreams of RMs that will be stored into external memory. So, the design configuration can be changed another configuration by replacing these RMs as shown in Fig. 4-5.

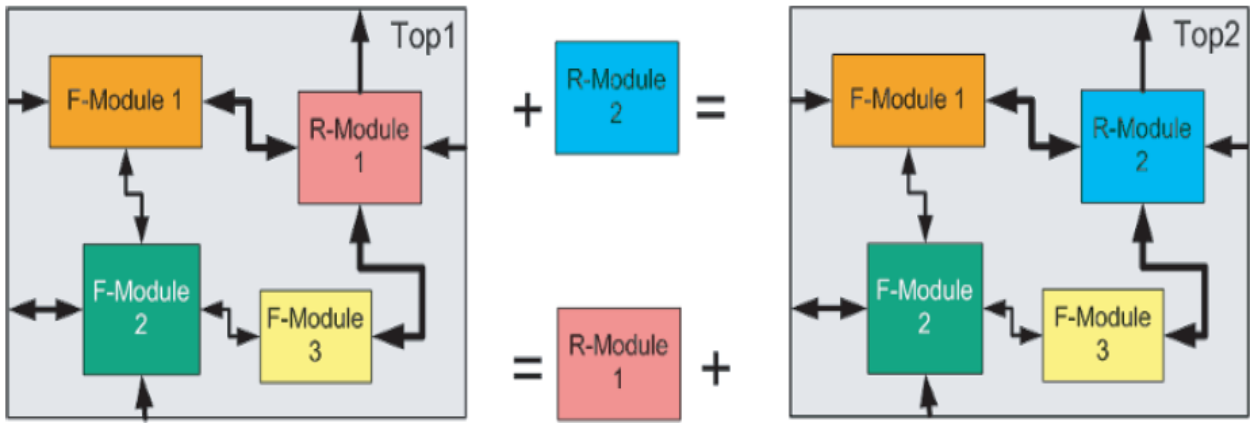


Figure 4-5. Module Based PR Configuration.

In this dissertation, the module-based PR design flow is used for the implementation of DPR section of DPRDS design.

4.3. FPGA Based BPSK and QPSK Demodulators

During the implementation of both BPSK and QPSK demodulators, which are the reconfigurable modules of DPRDS system, the designer has to take into consideration that the IO pins of both MPSK demodulators connected the MPSK demodulators with the static design must have the same number, name, type, and width to perform the DPR process properly.

4.3.1. BPSK Demodulator

The received signal is modulated by BPSK modulation and the classification decision informs the PRC IP core which demodulator has to be fetched from the external memory to reconfigure the dynamic section with. Then the BPSK demodulator is configured into FPGA, the received signal samples which are stored into FIFO buffer outputs to feed the BPSK demodulator input.

In Fig. 2-20, the carrier frequency has to be recovered from the modulated signal using Costas loop or PLL but for simplicity, a sinewave LUT is used instead. The samples of the received modulated signal are multiplied by the sinewave samples stored in LUT. The multiplication results are accumulated for the duration of one symbol (bit), and then a comparison to a threshold (zero) is made to determine either the received symbol represent bit 0 or bit 1, so the original binary sequence is recovered as seen in Fig. 4-6.

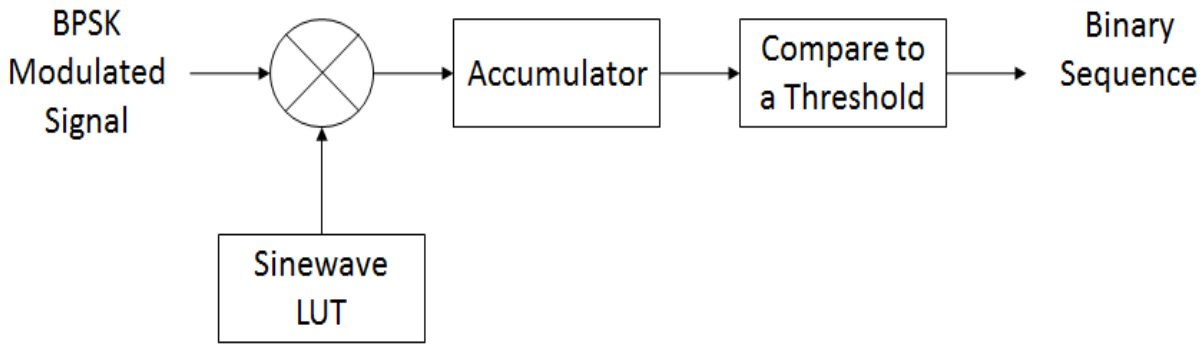


Figure 4-6. BPSK Demodulator.

4.3.2. QPSK Demodulator

The same sequence is performed to demodulate the received modulated signal as described for BPSK demodulation passing through the generation of classification decision that informs the PRC IP core where QPSK demodulator has to be fetched from the external memory to reconfigure the dynamic section with. After reconfiguring the FPGA by the QPSK demodulator, the received signal samples which are stored into FIFO buffer outputs to fed the QPSK demodulator input.

In Fig. 2-24, the carrier frequency has to be recovered from the modulated signal using Costas loop or PLL but for simplicity, a sinewave and cosinewave LUTs are used instead. QPSK demodulator considered as two BPSK demodulator called upper branch (I-ch) and lower branch (Q-ch). The samples of the received modulated signal are multiplied by the sinewave samples stored in LUT in upper branch and by cosinewave samples in lower branch. The multiplication results in both branches are accumulated for the duration of one symbol, and then a comparison to a threshold (zero) is made to determine either the values of I and Q in the upper and lower branches respectively, represented by bit 0 or bit 1. The resulting values are then multiplexing to recover the original binary sequence as shown in Fig. 4-7.

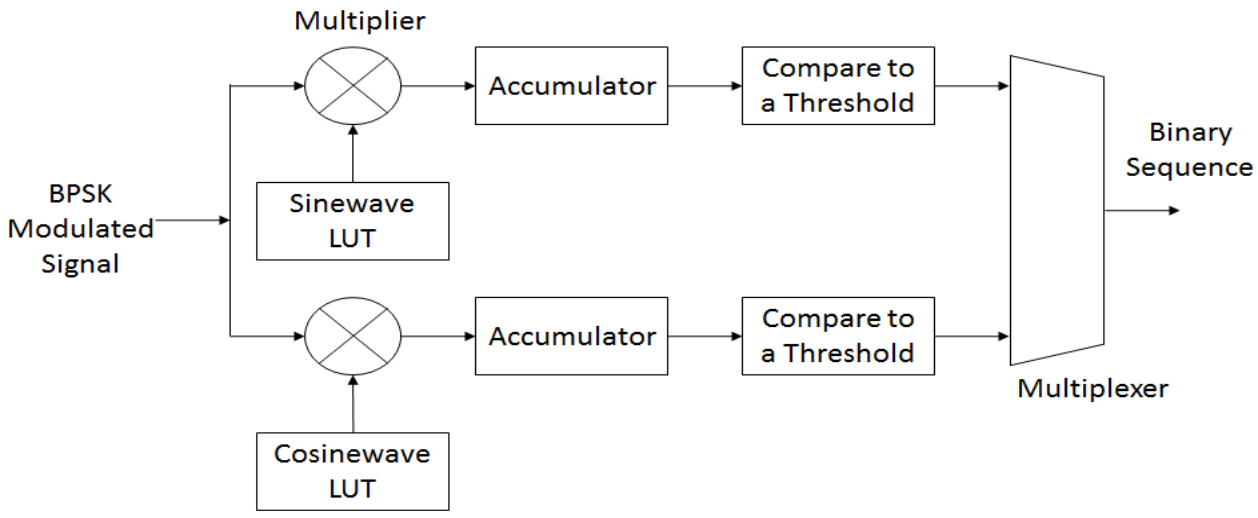


Figure 4-7. QPSK Demodulator.

4.4. FPGA Based DPRDS-DPR

The implementation of DPRDS-DPR part is required to define some aspects as; the suitable FPGA board that supports partial reconfiguration, how to generate full and partial configuration bitstream files, how the PRC will react according to the classification results and how it will manage the DPR process, and how the stored demodulator partial bitstream files (RMs) will be fetched from the external memory to reconfigure the dynamic section with it.

According to Xilinx user guide as shown in Table 4-2 [68], which states partial reconfiguration is supported using ICAP and the other configuration ports and it is supported also using Bus Peripheral Interface (BPI) at asynchronous mode only for Kintex-7 FPGA boards.

Table 4-2. Partial Reconfiguration Supported Configuration Ports

Configuration Mode	7 Series	Zynq	UltraScale	UltraScale+	Zynq UltraScale MPSoC
JTAG	Yes	Yes	Yes	Yes	Yes
ICAP	Yes	Yes	Yes	Yes	N/A
PCAP	N/A	Yes	N/A	N/A	Yes
MCAP	N/A	N/A	Yes	Yes	N/A
Slave Serial	Yes	N/A	Yes	Yes	N/A
Slave SelectMap	Yes	N/A	Yes	Yes	N/A
SPI (any width)*	No	N/A	No	No	N/A
BPI sync mode*	No	N/A	No	No	N/A
BPI async mode	Yes	N/A	Yes	Yes	N/A
Master modes	No	N/A	No	No	N/A

Kintex-7 kc7k160tfbg484-1 FPGA, which is used for the design and implementation of DPRDS-Classification part of the design, has a SPI flash memory which does not support DPR process. There is no problem of storing partial reconfiguration bitstream files into Serial Peripheral Interface (SPI) memory, but the problem appears during the DPR

process where “Startup” primitive that uses to manage the communication between the implemented design into FPGA and external SPI flash memory for disabling the clock of SPI flash memory at the beginning of DPR process and cannot deliver partial bitstreams to the configuration engine, therefore, DPR fails.

The other way to communicate with SPI flash memory without using “Startup” primitive is using the user IO pins, but the clock pin of SPI flash memory on Kintex-7 kc7k160tfbg484-1 FPGA device is not assigned as a user pin, which makes the using of this board for DPR process is difficult.

Kintex-7 KC705 evaluation board is used as the proper alternative solution to overcome the previous problem as recommended from Xilinx community forum. Kintex-7 KC705 has an asynchronous BPI flash memory which supports the implementation of DPR process without problem.

The controller of DPR process is the PRC IP core. PRC IP core consists of one or many virtual socket mangers connected to configuration memory and ICAP via a single fetch path. Through this path a bitstream stored on an external configuration memory is fetched and sent to the ICAP. PRC IP core considers the reconfigurable partition and any logic that exists in static design and support PR process as a virtual socket as seen in Fig. 4-8 [69].

Virtual Socket Manager is waiting for a trigger event to occur. A trigger can be hardware or software trigger. When a trigger occurs, the Virtual Socket Manager matches the trigger to a Reconfigurable Module (BPSK or QPSK demodulator) and manages the reconfiguration of that Reconfigurable Module.

Each virtual socket manager has two states; active and shutdown states. In active state, virtual socket manager controls a virtual socket, reposes to triggers, and loads RMs. In shutdown state, it does not control a virtual socket. So, it does not response to triggers or loads RMs.

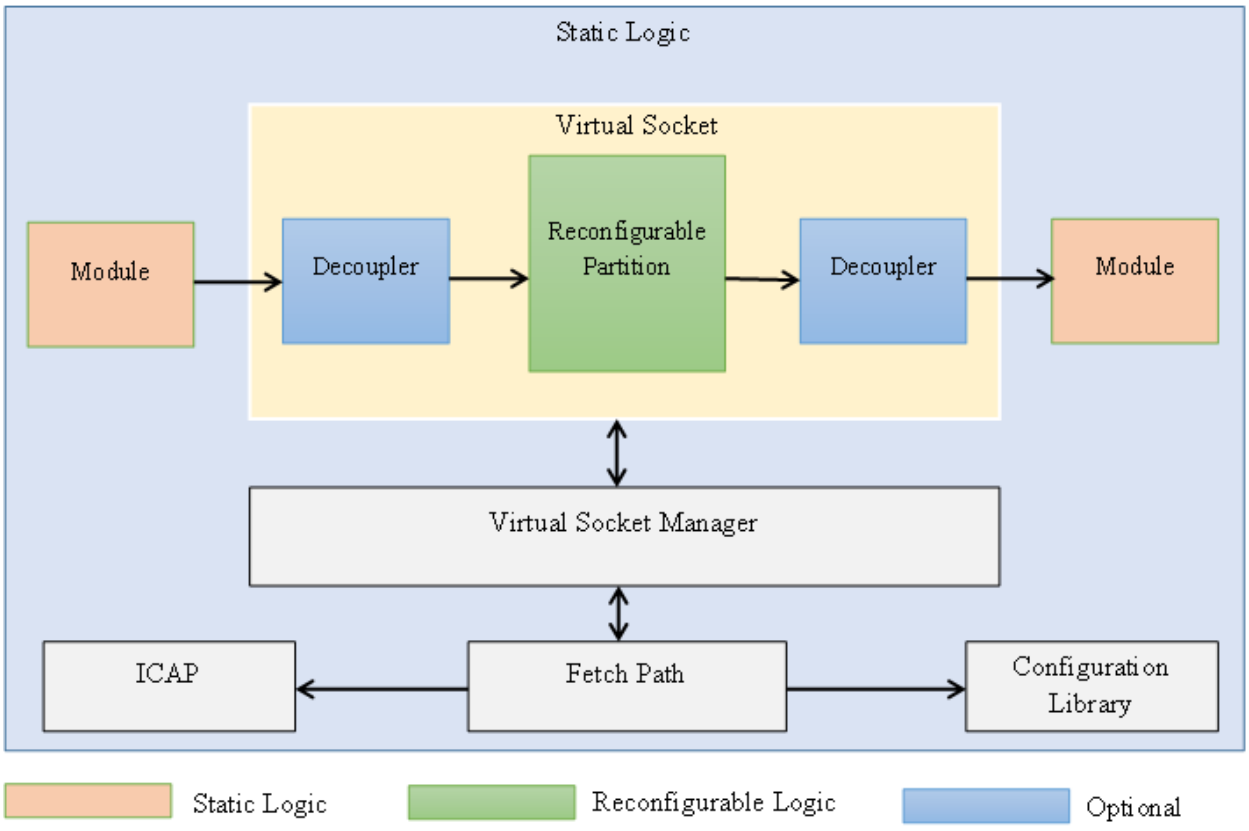


Figure 4-8. PRC IP Core Configuration.

PRC IP core decouples or isolates the static design modules from the reconfigurable partition during the reconfiguration process. During reconfiguration process the output data from a reconfigurable module is not valid until the completion of PR process and the reconfigure module is reset. This isolation is implemented into reconfigurable modules at the output ports using 2-1 multiplexer or by using enable signal to output data to static design after the reconfiguration process is completed. Xilinx provides PR Decoupler IP core to perform the decoupling process [68].

During the design of PRC IP core, the creation of virtual sockets and virtual socket managers is adjusted. The name, starting address, the size of a partial bitstream files, the order of these reconfigurable modules and its corresponding triggers, and other

parameters must be entered into a virtual socket (reconfigurable partition) as shown in Fig. 4-9.

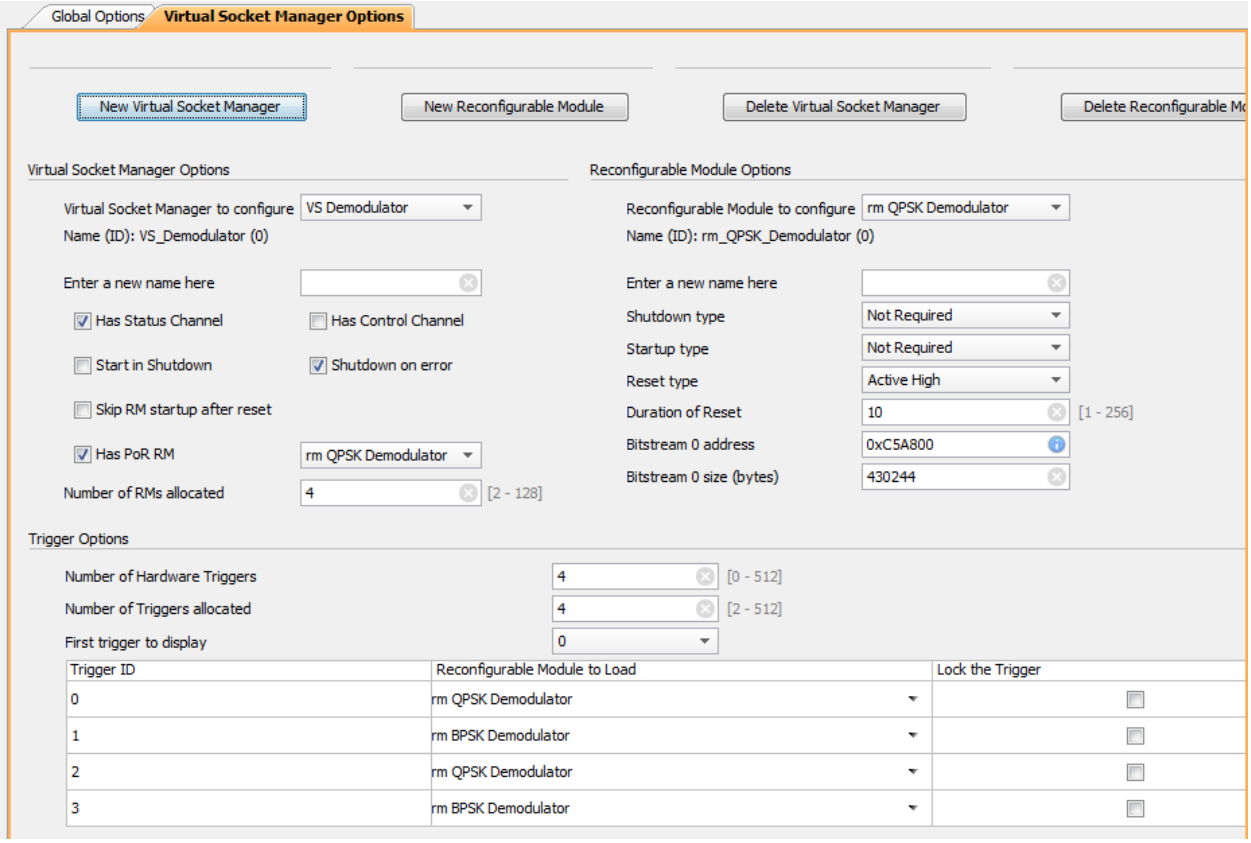


Figure 4-9. PRC IP Core Design.

When a trigger occurs, PRC IP core sends the address of the suitable partial bitstream file in external flash memory to EMC IP core to fetch the corresponding partial bitstream file (RM module) from external memory and sends it back to PRC IP core which in turns sends this file to ICAP interface to reconfigure the reconfigurable partition or dynamic section with the RM.

So the reconfiguration process is started when a trigger occurs, the existing RM is shutdown, then a new RM is loaded, and finally the new RM is reset as shown in Fig. 4-10 [69]. In the shutdown stage, The Virtual Socket Manager informs the RM that it will be removed, and waits until the RM gives permission.

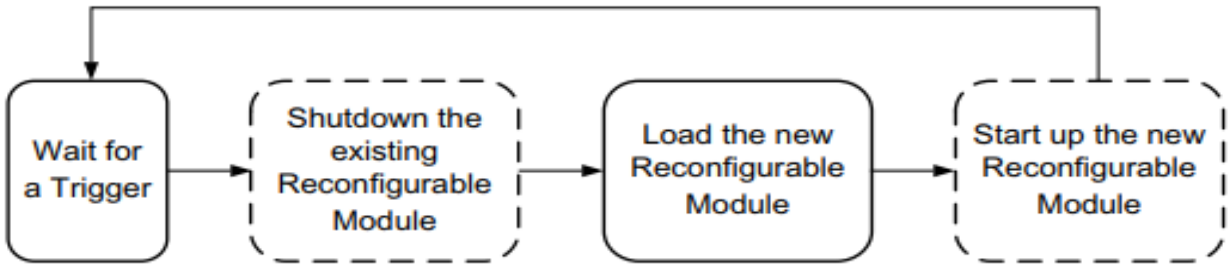


Figure 4-10. Virtual Socket Manager Steps during DPR.

PRC IP core stores the address in bytes, but the generated PROM image that will be stored into external flash memory stores address in 16 bits. So this matter has to be taken into consideration during the generation of PROM image [70].

4.4.1. Design and Implementation Flow

The DPRDS has to be implemented into two module groups; static and dynamic modules. The static modules consist of all the system modules except demodulator module which is a dynamic module. The static modules are fixed and will not be changed during runtime but demodulator module is not fixed and will be changed (BPSK or QPSK demodulator) during runtime without any influence on the rest of the design modules (static).

The property of demodulator module has to be changed to “out of context” before synthesizing the DPRDS design. This change will make VIVADO to synthesize demodulator module separately, where this synthesized file is needed during the implementation of DPR process in addition to the synthesized file of the static design.

After getting the synthesized files, a checkpoint is opened to set the “HD.RECONFIGURABLE” property of demodulator module and the location of this module on the FPGA floorplan. Then the property “Reset_After_Reconfiguration” is set to initialize RM after the completion of DPR process.

Implementation of whole DPRDS design for each demodulator (BPSK and QPSK) is performed by using a design constraints file. The output from this stage has two

checkpoint files (.dcp), and each one contains full implemented design with one of the demodulator schemes. To isolate the static design from dynamic design, the demodulator has to be replaced with blackboxes to generate an isolated static design.

The next stage is to generate full and partial bitstream files for the design which are used to create a PROM image that will be stored into the external flash memory (BPI Flash). A "write_cfgmem" command is used to create PROM image file. The image file contains the full bitstream, BPSK partial bitstream, QPSK partial bitstream, blockboxes partial bitstream, and their addresses on flash memory. A PROM image file is used to program BPI flash memory on Kintex-7 KC705 FPGA board. The design flowchart is as shown in Fig. 4-11.

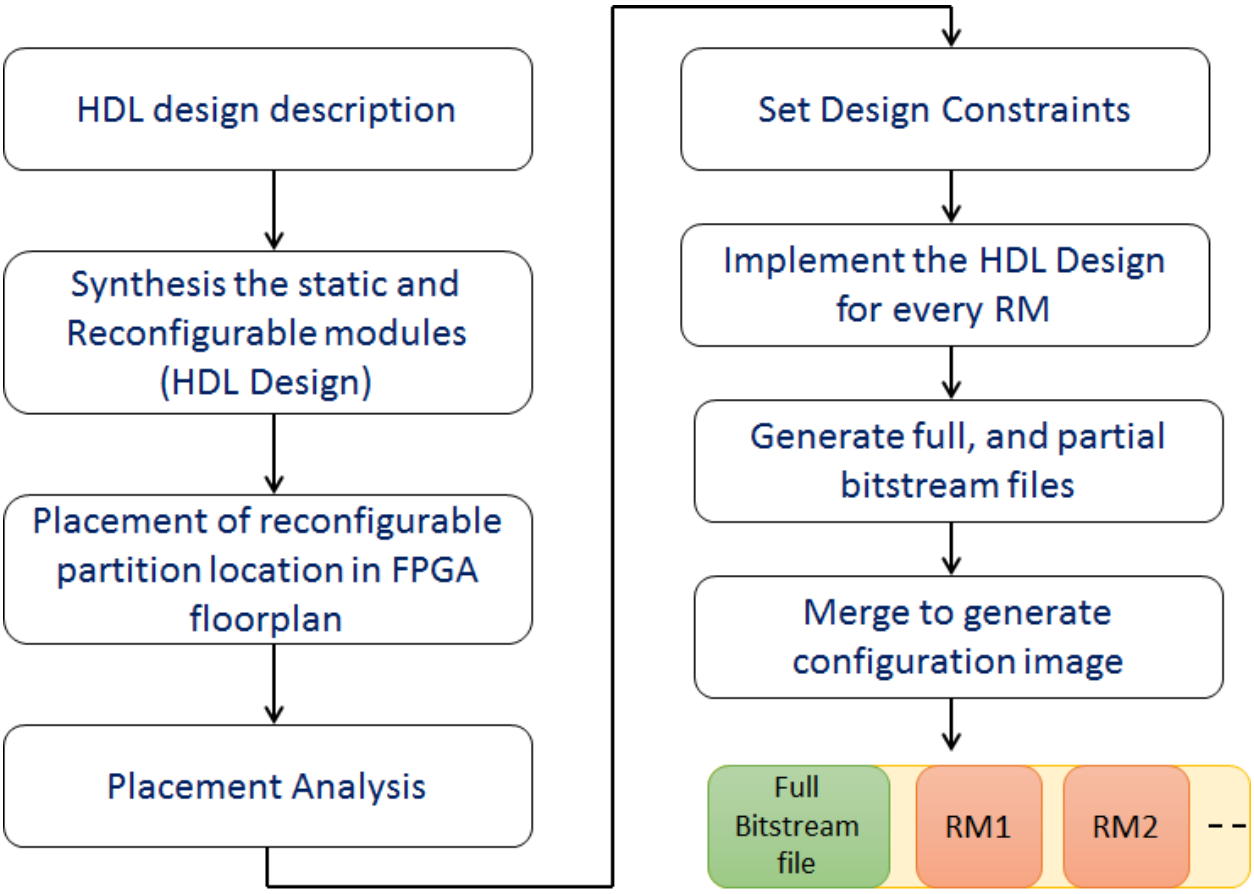


Figure 4-11. DPRDS Flowchart.

4.5. Performance Evaluation Aspects of DPRDS-Classification System

The evaluation of DPRDS-DPR part is performed using the BPSK and QPSK modulated signals generated from MPSK modulator as performed in DPRDS-Classification. The performance evaluation of DPRDS-DPR part has to be realized by calculating configuration and recognition time. The summarize of performance evaluation outlines are listed as follows;

- Determination of testing configuration of the system and how this system will be tested.
- Calculation of reconfiguration time required to reconfigure reconfigurable partition with corresponding reconfigurable demodulator modules.
- Comparing the reconfiguration time with that of the previous studies.
- Calculation of FPGA Power consumption and resources utilization of the realized system.
- Comparing resulting demodulated serial data with the original data.

4.5.1. Testing Configuration

Kintex-7 kc7k160tfbg484-1 and Kintex-7 KC705 evaluation boards are used to test the performance of DPRDS system. Kintex-7 kc7k160tfbg484-1 FPGA is used as MPSK modulator (BPSK and QPSK modulations) and a Kintex-7 KC705 is programmed by the whole design of DPRDS (for both classification and DPR processes).

A MPSK modulated signal, synchronous, and reset signals are used for the communication between both FPGA boards. The testing scenario starts with the generation of MPSK signal which is received by DPRDS system implemented on the other FPGA board.

The received MPSK signal passes through classification modules to determine the class or the modulation scheme of this signal. The classification decision (which monitored using LEDs) acts as a trigger to PRC IP core which matches the incoming trigger with the required partial bitstream file (modulation scheme). After that, PRC IP core sends the corresponding memory address and size of the required modulation scheme to the EMC IP core.

EMC IP core in turns fetches the required partial bitstream file which is already stored into external BPI flash memory then sends this file to PRC IP core that sends it to ICAP port to reconfigure dynamic section (reconfiguration partition) by the proper demodulator. Two user LEDs are used as indicators of DPR process together with ILA. The testing configuration of DPRDS system is shown in Fig. 4-12.

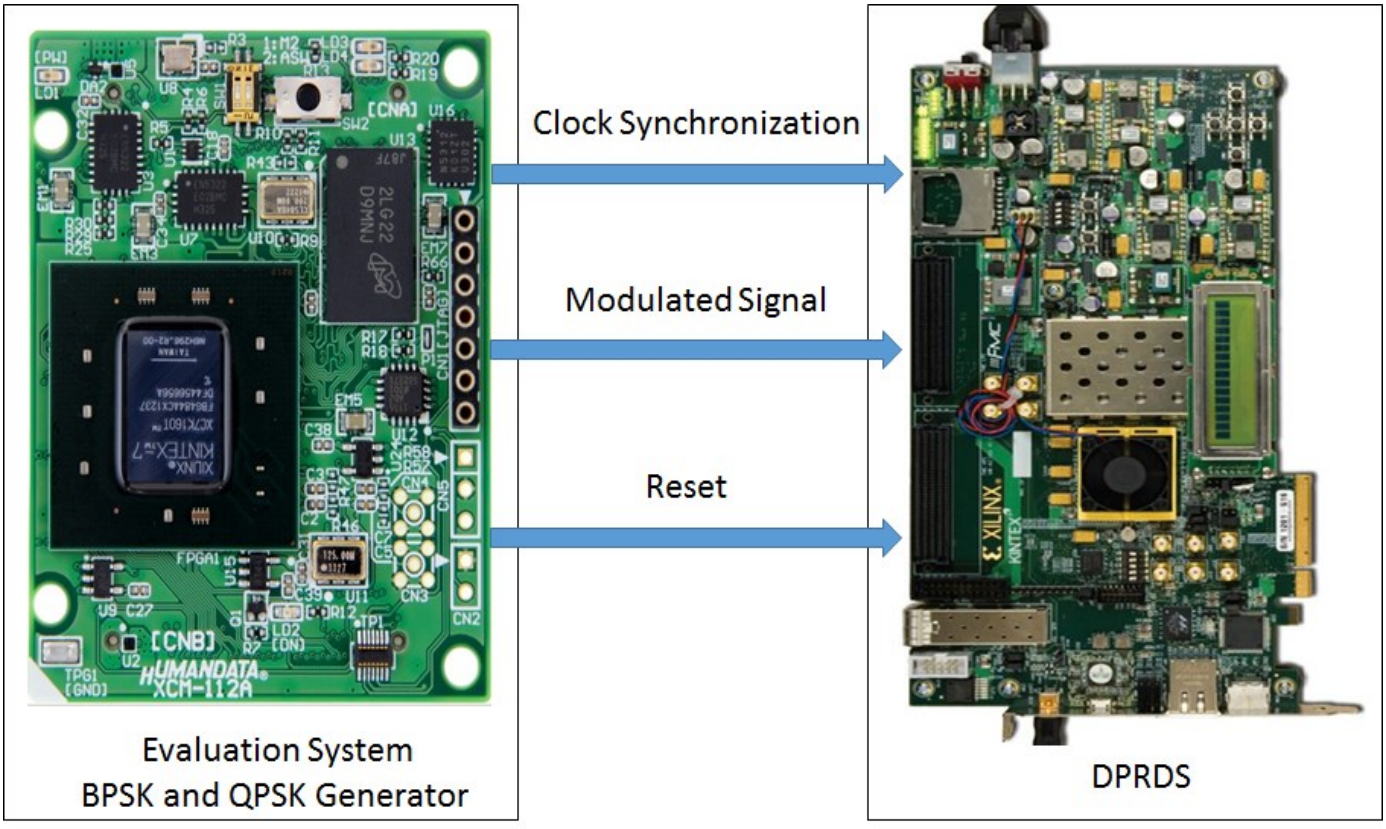


Figure 4-12. DPRDS Testing Configuration.

4.5.2. Results and Discussion

In this section, the steps of the DPR process are described and the resulting runtime output using ILA is shown and discussed. The design flow of DPRDS system is described in Fig. 4-11, and the most important factor that makes the DPR process success is the precise preparation of that process. The demodulator module (RM) for both BPSK and QPSK modulation schemes must be synthesized separately from the whole design by changing the property of demodulator module to be “out of the context”.

The synthesis process is intended to represent the RM or demodulator module as a blackbox into the static design, which is loaded with the synthesized file of demodulator module during the implementation of the whole design. The resulting synthesized files from the synthesis process are three files (.dcp); static design with blackbox, BPSK, and QPSK synthesized files.

The static synthesized file is the file that DPR process starts, a check point is opened by using static synthesized file, and then the BPSK synthesized file is loaded. Next step is to locate the BPSK demodulator module at specific location on FPGA floorplan to generate a reconfigurable partition. Then, the “HD.RECONFIGURABLE” and “RESET_AFTER_RECONFIGURATION” properties have to be set to be true.

A design placement verification must be performed to be sure there is no error while determining the location of reconfigurable partition. Then the constraints file is loaded to design for the implementation of the design using BPSK demodulator. The generation of bitstream files is performed after the implementation, the resulting bitstream files are; full bitstream, and BPSK partial bitstream files.

Then the BPSK demodulator module is removed from the design and a generation of bitstream is occurred to generate blackbox partial bitstream file that is added to the final PROM image. In this step, the design can add the QPSK demodulator instead of the generated blackbox to generate full and QPSK partial bitstream files without the repetition of the previous steps.

The BPSK and QPSK are converted to bin files before adding these files to full bitstream file to create PROM image to be loaded into the external BPI flash memory. The steps of DPR implementation is shown in Fig 4-13.

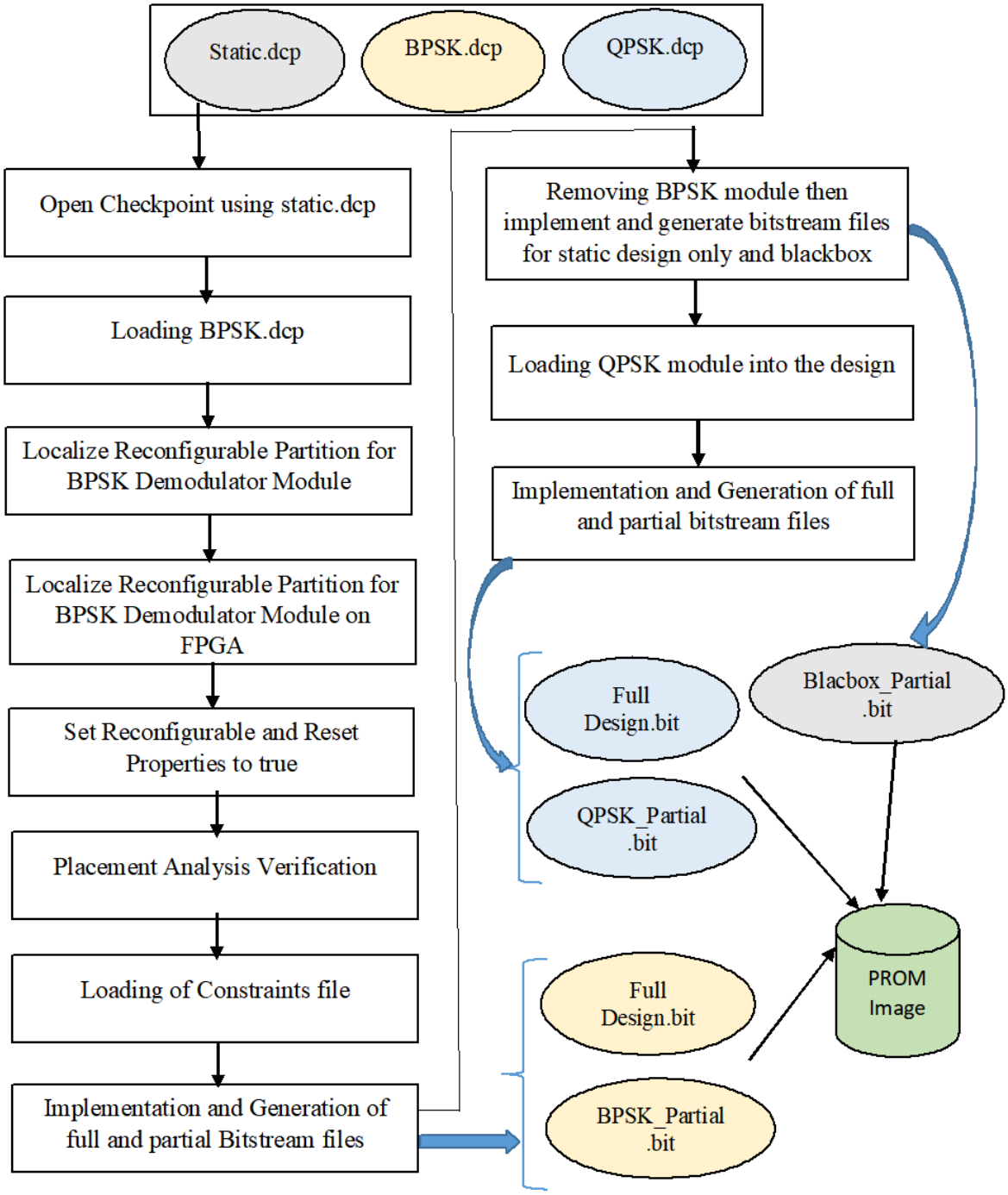


Figure 4-13. DPR Implementation Steps.

The floorplan of Kintex-7 KC705 evaluation board before and after the implementation of BPSK/QPSK demodulator module into reconfigurable partition is as shown in Fig. 4-14 and 4-15.

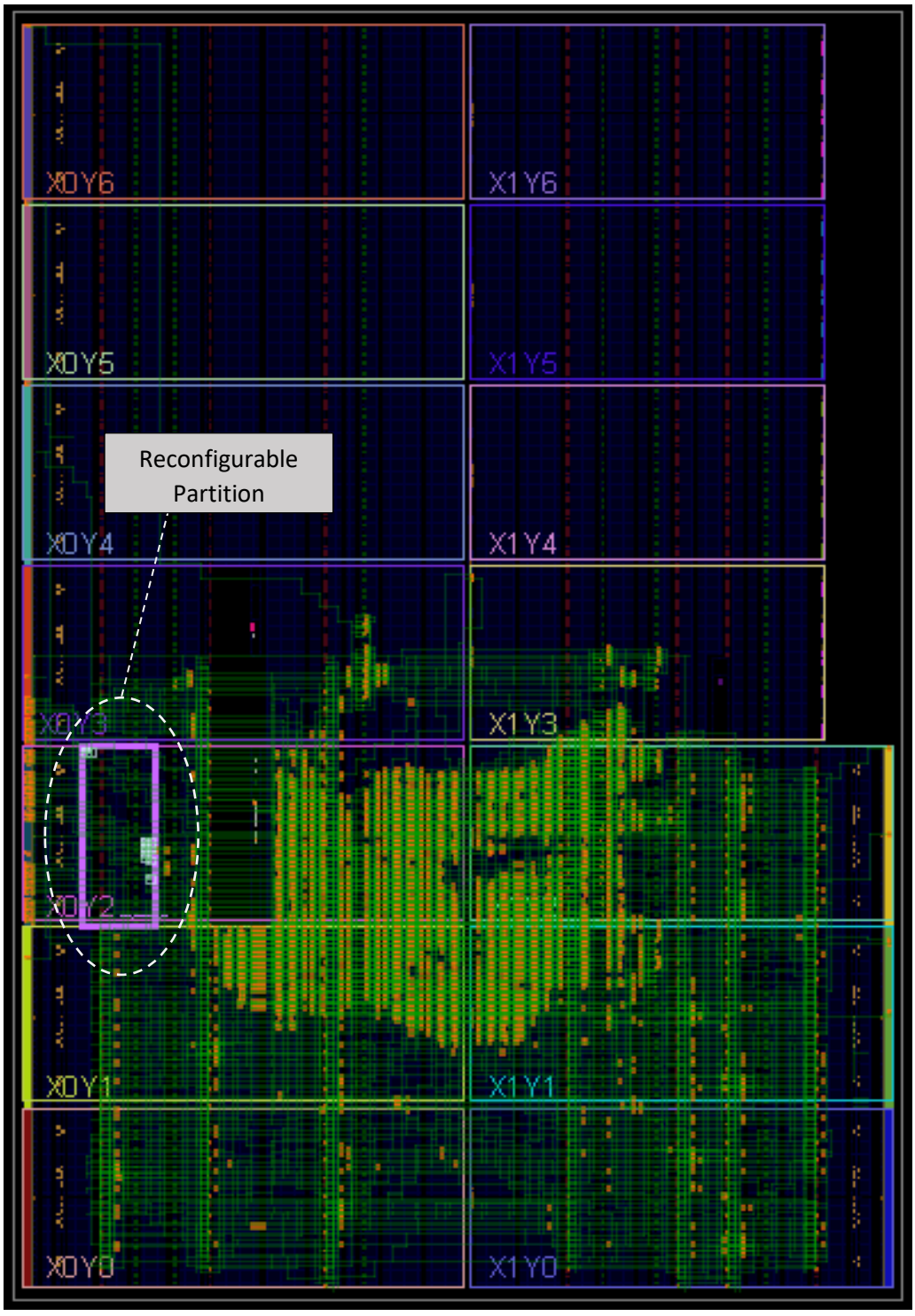


Figure 4-14. DPRDS FPGA Floorplan without Demodulator Module.

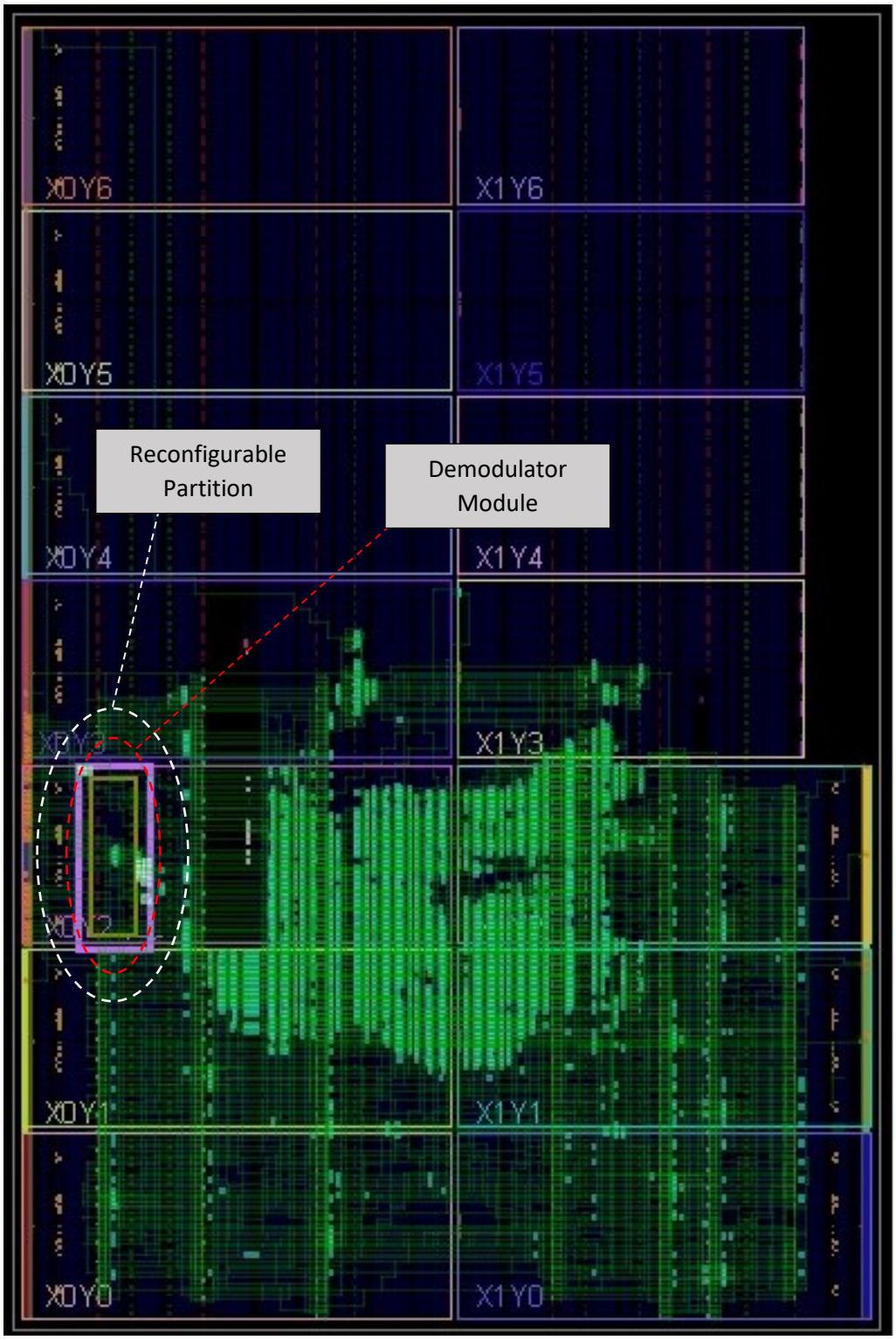


Figure 4-15. DPRDS FPGA Floorplan with Demodulator Module.

The Partition pins located at the boundaries of reconfigurable partition connect the static design and the DPR design together after the reconfiguration process. The Partition pins are the white squares that appears in FPGA floorplan as shown in Fig. 4-16.

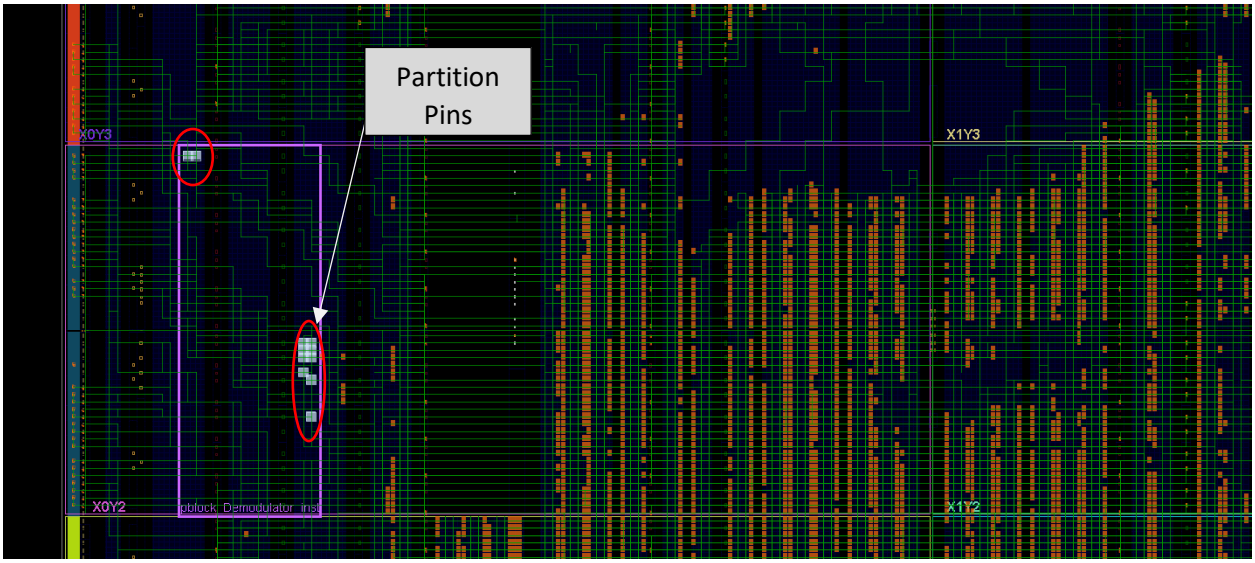


Figure 4-16. Partition Pins.

A PROM image that contains full bitstream file with BPSK demodulator, and BPSK and QPSK partial bitstream files are used to test the DPRDS system. At the beginning, a QPSK modulated signal is sent to the DPRDS system that is loaded at the FPGA startup from BPI flash memory. In that case there is no trigger occurs because the received signal and demodulator are QPSK and the resulting demodulated data is coincided with the original binary data.

Changing the user switch to generate a BPSK modulated signal stimulates the classifier to generate a BPSK trigger which informs PRC IP core by the targeted demodulator. PRC IP core performs the fetching process of BPSK demodulator configuration file, using the corresponding address stored before, from BPI flash memory by using EMC IP core. after that, PRC IP core sends this configuration file to ICAP port to reconfigure the reconfigurable partition with BPSK demodulator. The resulting binary data sequence is coincided with the original serial binary data.

Changing the user switch to generate a QPSK modulated signal stimulates the classifier to generate a QPSK trigger which informs PRC IP core by the targeted demodulator. PRC IP core performs the same steps in the previous paragraph till the reconfiguration of the reconfigurable partition with QPSK demodulator. The resulting binary data sequence is coincided with the original serial binary data and so on.

The resulting output when a BPSK trigger occurs is shown in Fig. 4-17 using ILA during runtime.

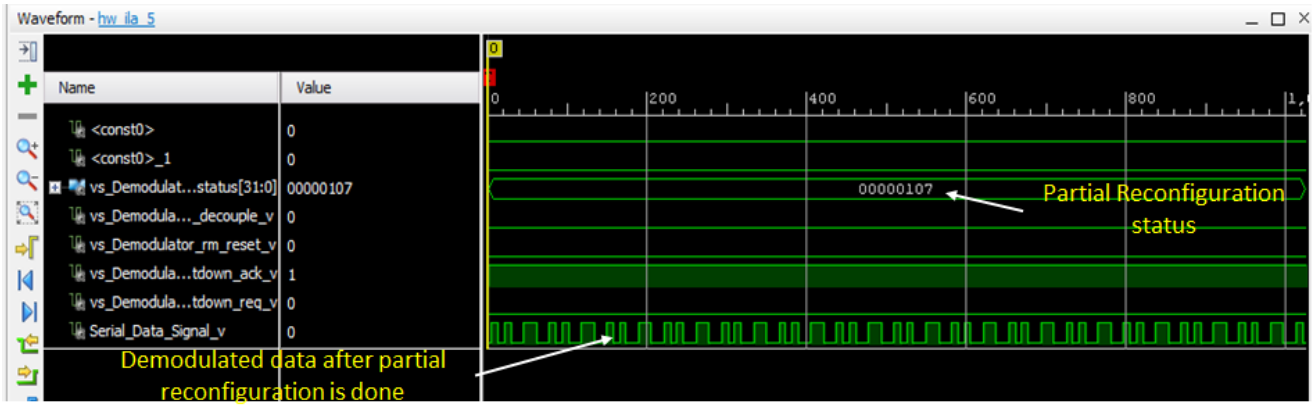


Figure 4-17. Resulting Output after Reconfiguring the Reconfigurable Partition with BPSK Demodulator.

Virtual socket status or partial reconfiguration status is used during runtime to indicate which reconfigurable module is loaded into reconfigurable partition. The resulting value of partial reconfiguration register is described as follows;

Virtual Socket Status = 00000107 → 0000_0000_0000_0001_0000_0111

- RM_ID (bits 23:8) = 1. This means RM 1 (BPSK Demodulator) is loaded.
- SHUTDOWN (bit 7) = 0. This VSM is not in the shutdown state.
- ERROR (bits 6:3) = 0000. There are no errors.
- STATE (bits 2:0) = 111. The Virtual Socket is full.

The status means that the BPSK demodulator is loaded successfully. Also the demodulator output binary data proves that the demodulator output is coincided with the original binary data.

The same situation for QPSK demodulation output when a QPSK trigger occurs as shown in Fig. 4-18.

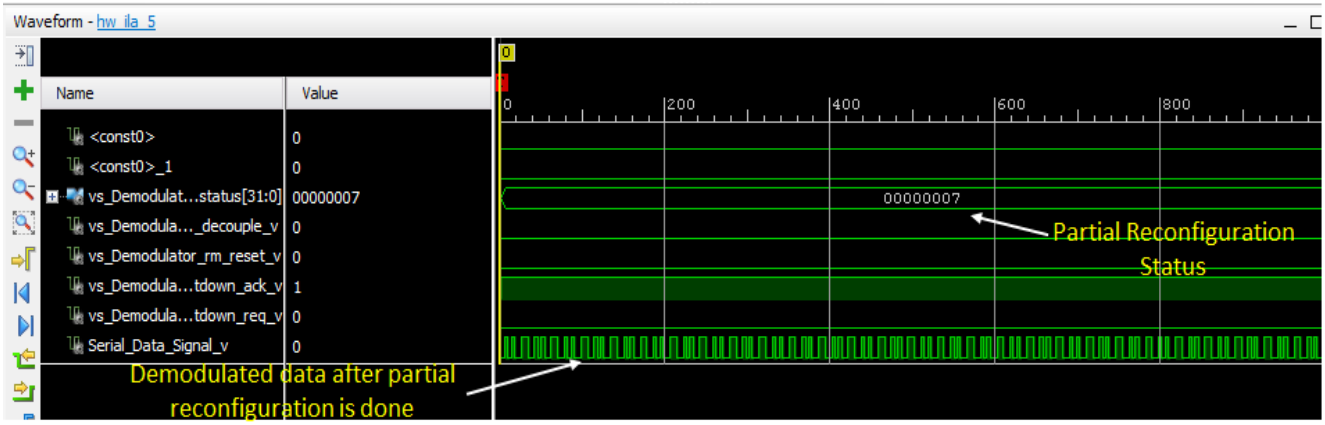


Figure 4-18. Resulting Output after Reconfiguring The Reconfigurable Partition with QPSK Demodulator.

Virtual Socket Status = 00000007 → 0000_0000_0000_0000_0000_0111

- RM_ID (bits 23:8) = 0. This means RM 1 (BPSK Demodulator) is loaded.
- SHUTDOWN (bit 7) = 0. This VSM is not in the shutdown state.
- ERROR (bits 6:3) = 0000. There are no errors.
- STATE (bits 2:0) = 111. The Virtual Socket is full.

The status means that the QPSK demodulator is loaded successfully. Also the demodulator output binary data proves that the demodulator output is coincided with the original binary data.

These results show that the DPR process is working properly according to the classification decision. By other meaning, the DPRDS system is working properly as whole.

The configuration time of the reconfigurable module is based on the size of reconfigurable module and the maximum bandwidth of the configuration port. The reconfiguration time can be computed using Eq. (4-1).

$$Configuration\ Time\ (CT) = \frac{\text{Partial bit−stream file size}}{\text{Configuration Port maximum bandwidth}} \quad (4\text{-}1)$$

the size of BPSK and QPSK partial bitstream file is 430244 bytes, and maximum bandwidth of ICAP port is 3.2 Gbps as shown in Table 4-1. So, the configuration time is computed as follows;

$$Configuration\ Time\ (CT) = \frac{430{,}244 * 8}{3.2 * 10^9} = 1.1\ msec$$

DPRDS design schematic is as shown in Fig. 4-19.

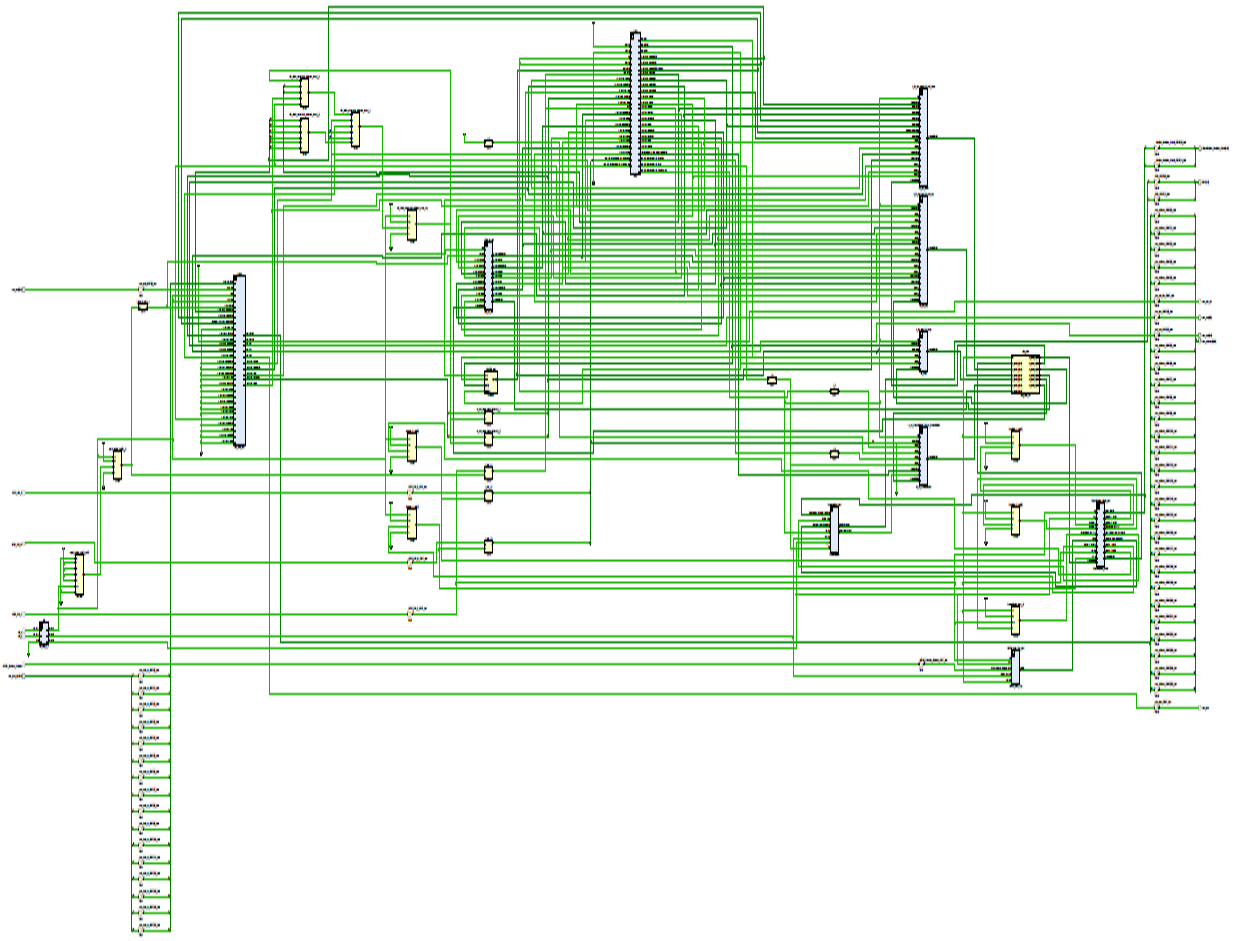


Figure 4-19. Schematic of DPRDS Design.

One of the objectives of this research is to design and implement DPRDS system with less resources utilization and low power consumption. Using MATLAB in collaboration with VIVADO let the design of this system be less complex by implement the classifier training stage using MATLAB and just using the resulting parameters for the design of the classifier classification stage, getting wavelet filter coefficients, and getting some output results using MATLAB.

The resources utilization and power consumption are shown in Table. 4-3 and Fig. 4-20 respectively.

Table 4-3. Kintex-7 FPGA Resources Utilization for DPRDS System

Resource	Utilization	Available	Utilization%
LUT	10,246	203,800	5.03
LUTRAM	1,861	64,000	2.91
FF	14,598	407,600	3.58
BRAM	206.50	445	46.40
DSP	42	840	5.00
IO	57	500	11.40
BUFG	6	32	18.75
MMCM	1	10	10.00

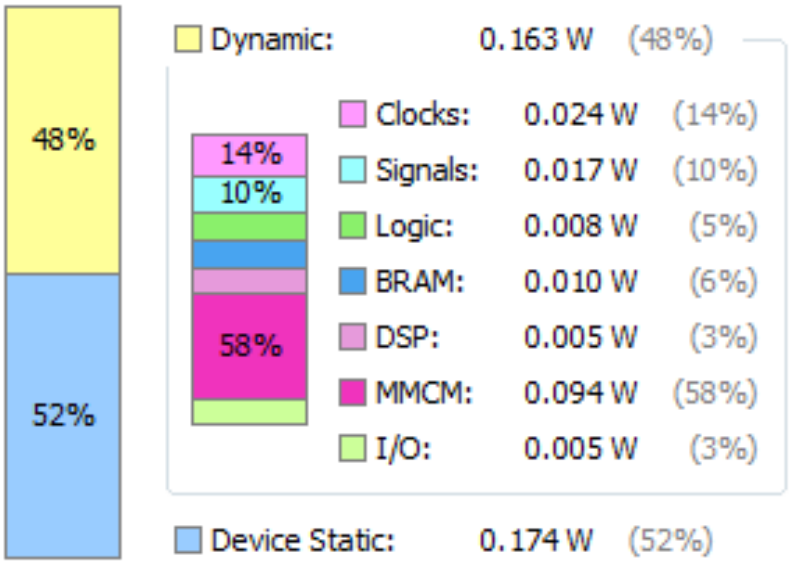


Figure 4-20. Power Consumption of DPRDS System.

The total power consumption of the DPRDS system is the summation on static and dynamic power consumption. So, the total power consumption of DPRDS is equal to 0.337 W.

This chapter described the DPR methodology, design, and implementation of DPRDS system. The DPR is described in detail in the help of figures and flowcharts and runtime output results from ILA. The configuration time, resources utilization and power consumption are also described. The configuration time is about 1.1 msec with less resources utilization and low power consumption.

Next chapter describes the ability of sharing ICAP port among DPRDS and other applications that need the access to ICAP without any influences on the performance of these applications. As example, SEU mitigation application is used with the presence of DPRDS system to test the availability of multiple access of ICAP port.

Chapter 5 : ICAP Multiple Access by DPRDS and SEU Mitigation Systems

In this chapter, multiple access of Internal Configuration Access Port (ICAP) by Dynamic Partial Reconfigurable Demodulation System (DPRDS) and Single Event Upset (SEU) mitigation system is described. This chapter has two goals; first goal, is to show the possibility of sharing ICAP primitive or port among user applications which are DPRDS and SEU mitigation system in this chapter. The second goal is to reduce the injection time of fault errors (single events) into DPRDS.

5.1. Background of SEU mitigation

SEU or soft error is an unintended change to the state of a configuration memory bit caused by ionizing radiation which causes unacceptable changes in the design behavior. SRAM-based FPGAs are sensitive to ionizing radiation effects. Ionizing radiation alters the value of FPGA configuration memory (bit/bits) and causes unexpected behavior of implemented design which is called SEU. SEU shows a high probability of error in applications implemented on FPGAs [71].

There are several methods to mitigate the effects of ionizing radiation. A fault injection emulator system based on partial reconfiguration and triple Modular Redundancy (TMR) that injects random faults in FPGA configuration memory can be used to reduce the effects of radiation [72], [73], [74]. A dynamic partial reconfiguration-based fault-injection platform (DPR-FIP) can be used to emulate the SEU faults in FPGA configuration memory, it has the ability to cumulative SEU and supports both multi-event upset and single event faults in flip-flops and combinational parts [75].

Validating fault emulation systems by comparing the deployed results and accelerated test results against fault emulation improves the ability to accurately model the effect of upsets effects within FPGAs and increase the ability to identify problems that can be occurred in fault emulation system [76].

Implementation of a low area-overhead SEU recovery mechanism using Frame Error Correcting Codes (ECC) and Finite State Machine (FSM) increase the reliability of fault injection system and enable this system to inject errors in specific location into configuration memory [77]. Reduction of area overhead is important as the reduction of the cost. Low cost fault injection system based on checkpointing and CRC achieves high mitigation reliability [78]. Scan and scrubbing of FPGA configuration memory as fault detection and repair mechanisms as well as, rollback and checkpointing as fault recovery both of them can be used to implement low-cost soft error mitigation system [79], [80].

Principle of SEU-recovery techniques or configuration scrubbing is to use the FPGA configuration interface, as ICAP interface, to recover the original state of configuration memory and can be used for full or partial reconfiguration of configuration memory [81]. Prediction of error rate of any implemented design by combining FPGA static cross-section with the results of fault injection campaigns reduces the number of errors [82]. Injecting multiple errors at the same time (pipelining) into configuration memory increase the speed of the error injection process and the needed time is shortened [83]. Xilinx provides Soft Error Mitigation (SEM) IP core to emulate the effects of ionizing radiation in laboratory. SEM IP core can inject, detect, classify and correct errors in configuration memory.

This chapter describes a method that both soft error mitigation and partial reconfiguration processes can share ICAP primitive during runtime especially SEM IP core and PRC IP core cannot access ICAP primitive at the same time.

In addition, it describes a fault errors injection system that reduces the time needed to inject errors in both partial and non-partial regions of FPGA during runtime using a new method based on Xilinx essential bits technology.

The soft error injection and mitigation principle and the soft error mitigation system are described in the following sections.

5.2. The Principle of Soft Error Injection and Mitigation System

Error injection using SEM IP core is a low-cost method that uses to emulate the effects of ionizing radiation on the design behavior implemented on FPGA. Ionizing radiation alters the value of bit/bits of configuration frames which affects the behavior of the design. Xilinx provides SEM IP core that can inject errors in configuration frames and mitigate these errors depending on the mode selected by the user. SEM IP core uses essential bits technology to reduce the Effective Failures in Time (FIT), which is the number of design failures that can be expected in 109 hours, and to increase the availability of the design. For example, FIT for Kentix-7 xc7k325tffg900-2 is 7.2 FIT [84].

SEM IP core can be used for error injection only or for error injection, detection, classification and correction modes, and can be used for error injection and correction depending on the selected mode of the core. In this paper, error injection and correction mode is used to test the performance of sharing ICAP primitive between partial reconfiguration and soft error mitigation processes, and to emulate the effects of ionizing radiation on both static and dynamic sections of the partial reconfiguration system within minimum time using Xilinx essential bits technology.

The system consists of Kintex-7 xc7k325tffg900-2 (KC705) FPGA where Design Under Test (DUT) or Dynamic Partial Reconfigurable Demodulation System (DPRDS) is implemented and the fault injection is applied, and a PC runs MATLAB program to control SEM IP core, in order to control and monitor the error injection and mitigation processes as shown in Fig. 5-1. SEM IP core mitigates SEU and Multiple Bit Upset (MBU). MBU means that SEM IP core mitigates two or more error bits occur in one frame.

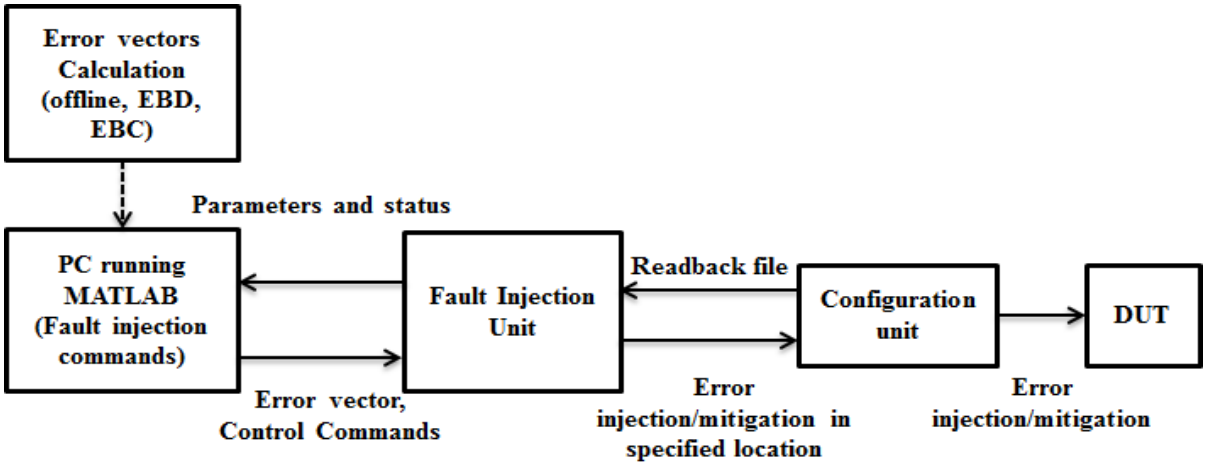


Figure 5-1. Error Injection and Mitigation Principle.

The default state of SEM IP core is “Observation”. So, after a full bitstream is loaded into a device, SEM IP calculates the golden frame ECC values and the device-level CRC value for the current configuration memory. Observation state means SEM IP core scans for errors in the configuration memory according to ECC and CRC values. Before performing Partial Reconfiguration (PR), and MATLAB program sends a command to SEM IP core to change the state to “IDLE”. When PR is performed, the configuration memory is changed, and then MATLAB program sends a soft reset command to SEM IP to recalculate the golden frame ECC values and the device-level CRC value to prevent false errors from being detected after PR is performed. After PR completes, MATLAB program sends a command to SEM IP to change the state to be “Observation” where it scans the entire configuration memory for errors based on the PR that occurred.

A command is sent to SEM controller to change the state to be “IDLE” then errors vector sends to SEM controller to inject errors in PR and non-PR regions of the device. The errors are detected and corrected by the SEM controller when it is transitioned to the “Observation” state to resume scanning.

The error injection rate is obtained from CREME96 model within the Space Environment Information System (SPENVIS) website [85] for orbit altitude of 614 km and elevation angle of 97.8º.

5.3. Sharing ICAP for Soft Error Mitigation and Partial Reconfiguration

Injection of errors and errors mitigation using SEM IP in both static and partial reconfigurable regions in FPGA requires sharing ICAP primitive between DPRDS and SEM IP [16]. ICAP is a primitive that uses to reconfigure the dynamic section of FPGA with the suitable RM in case of partial reconfiguration and to inject and mitigate errors in case of soft error mitigation. Either PRC IP core of DPRDS or SEM IP core can access ICAP primitive at the same time. The simple way to solve this problem is to use a multiplexer. Multiplexer allows the PRC IP core of DPRDS to access ICAP during partial reconfiguration process after that return control to SEM IP to continue soft error mitigation processes as shown in Fig. 5-2.

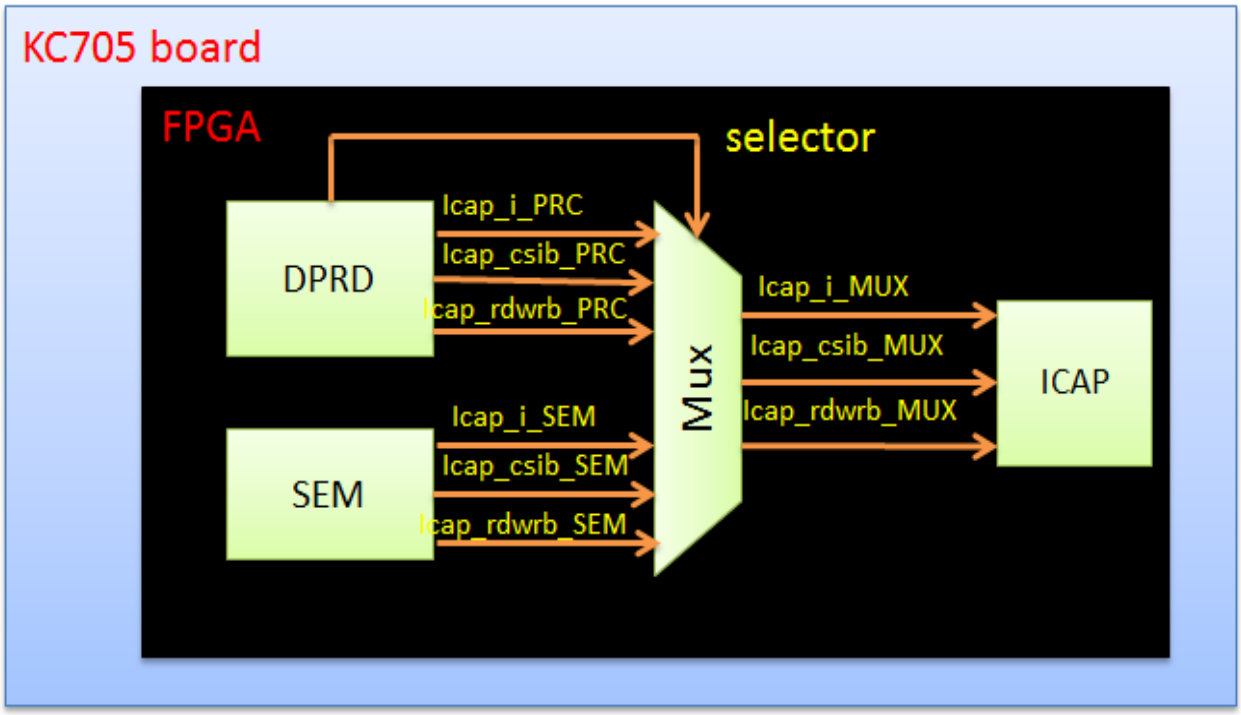


Figure 5-2. Sharing ICAP Primitive Concept

If the decision value of the classifier module is changed, the control of ICAP primitive is transferred to DPRDS and stay there till the partial reconfiguration process finish. During partial reconfiguration process, SEM IP core is in “Idle” state to prevent SEM IP from count and mitigate the new configuration as errors in FPGA configuration file.

The control of ICAP primitive returns to SEM IP after the end of partial reconfiguration process. A soft rest is sending to SEM IP core to recalculate golden frame ECC and CRC. Then fault injection process in partial reconfiguration section is performed when SEM IP state is “Idle”. After injection of errors in partial reconfiguration section, the state of SEM IP changes to “Observation” state to detect and mitigate the injected errors.

5.4. Soft Error Mitigation System

Soft error mitigation system architecture is as seen in Fig. 5-3. SEM IP core is Xilinx IP core used for soft error/fault injection and at the same time used for error mitigation. SEM IP core can be controlled form outside FPGA to allow users to emulate fault injection process. In this paper, SEM IP core is controlled from a PC running MATLAB program through UART serial communication interface. Injection of errors change states of SEM IP core, and all communication commands are done via UART interface.

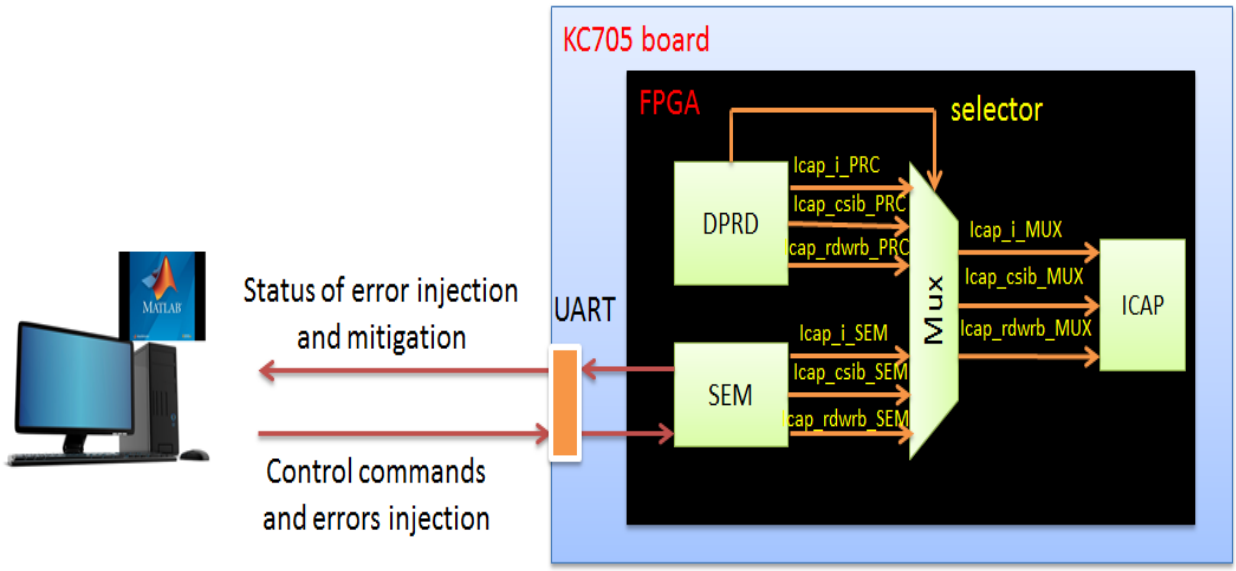


Figure 5-3. Soft Error Mitigation System Block Diagram

soft error mitigation system has to inject fault or errors in both static and dynamic partitions. For each FPGA chip there is a maximum frame number of the configuration file. The maximum number of configuration frames of Kintex-7 xc7k325tffg900-2 FPGA is 23,265 and each frame has 101 words each word contains 32 bits. In past, it is required to inject errors in all configuration cells/frames of FPGA device which consumes much time and reduces system availability and reliability standards However, there is no need to inject errors in all frames bits in the configuration file of FPGA because only a part of the total number of configurable memory cells are used [17]. The memory bits of unused cells are not essential to the design; instead it is just required to know the address of the memory bits of used cells or as called essential bits.

A property “essential bits” has to be set to “yes” into the constraint file before the generation of bitstream file of the design. This property allows generation of some files for static and each dynamic section. These files are Essential Bits Configuration (EBC) and Essential Bits Definition (EBD) files. The EBC file contains device configuration cells contents. The EBD file contains mask data that indicates which bits of the EBC file are essential to the circuitry of the design. In other meaning, a 1 in EBD file corresponds to an essential bit in EBC file. The logical addresses of these essential bits are calculated and formatted into fault injection vector. The injection rate of errors is determined depends on SEU rate which is obtained from CREME96 model within SPENVIS online package.

The emulation of radiation effects is performed by the injection of errors or changing the state of essential bits randomly for each frame that has essential bit/bits for both static and dynamic sections of FPGA design. The state of SEM IP core must be changed from “Observation” to “Idle” before insertion of errors and return back to “Observation” after the insertion of errors to mitigate these errors. Before the start of emulation test, configuration frames logical addresses of essential bits must be determined for both static and dynamic sections of the design as mentioned before by comparing the binary values

of EBD and EBC files of static and dynamic sections. The flowchart of the sequence of fault injection and mitigation process is described in a flowchart as shown in Fig. 5-4.

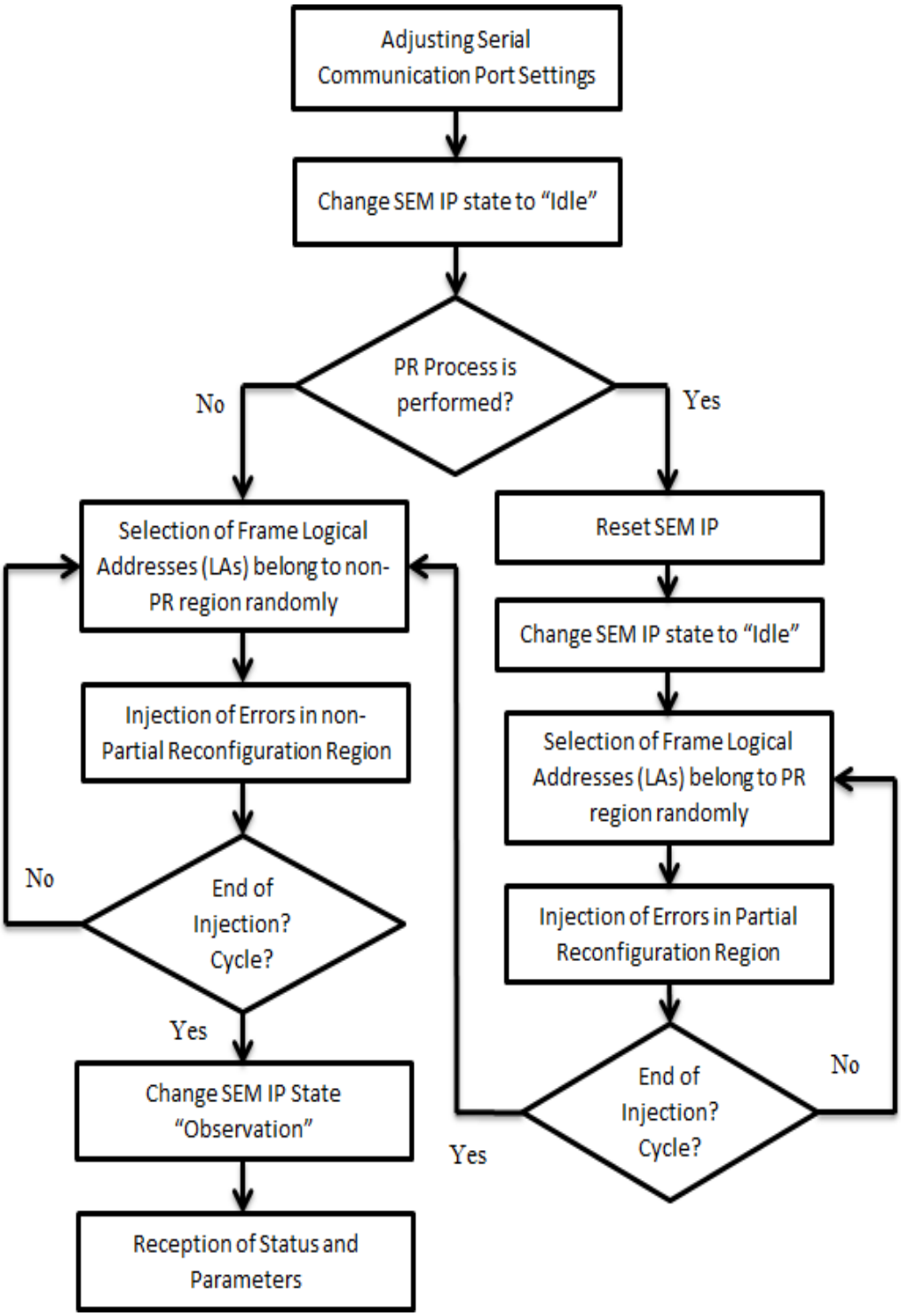


Figure 5-4. Error Injection and Mitigation Flowchart.

5.5. Results and Discussion

Fault/error injection and mitigation system is applied on DPRD design to describe a method that reduces the required time to inject errors in both partial and non-partial regions of FPGA. To achieve that, it is required to inject errors in essential bits of configuration memory and to inject errors in both partial and non-partial regions of FPGA and at the same time perform partial reconfiguration process it is required to share ICAP primitive between error injection and partial reconfiguration processes. Fig. 5-5 shows floorplan of the implemented design.

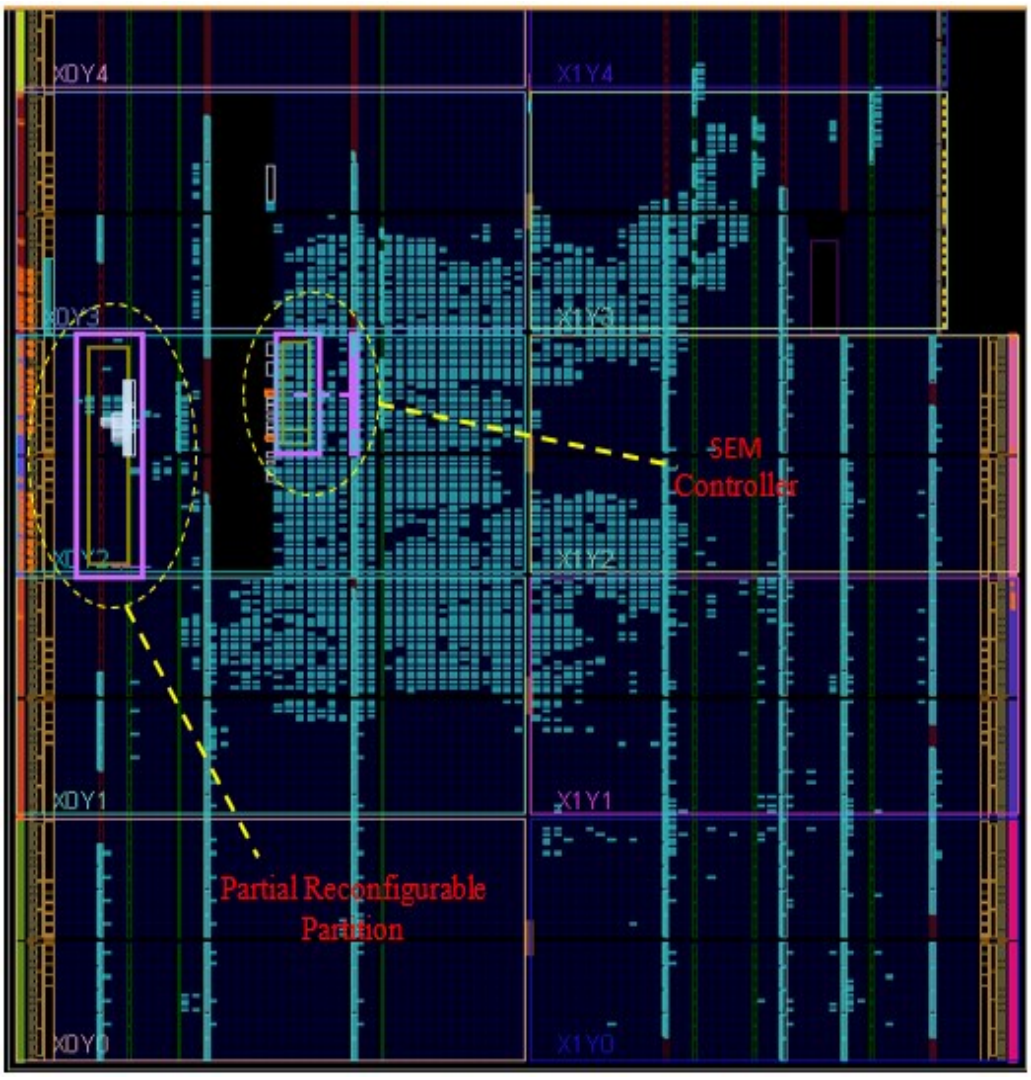


Figure 5-5. FPGA Floorplan of Full Design.

For partial reconfiguration process, the simulation results in Fig. 5-6 shows that when the classifier decision value changes the control of the ICAP is transferred successfully to PRC IP core to perform partial reconfiguration process of corresponding RM. The control of ICAP is reserved to PRC IP core until the end of partial reconfiguration process for the return back to SEM IP core.

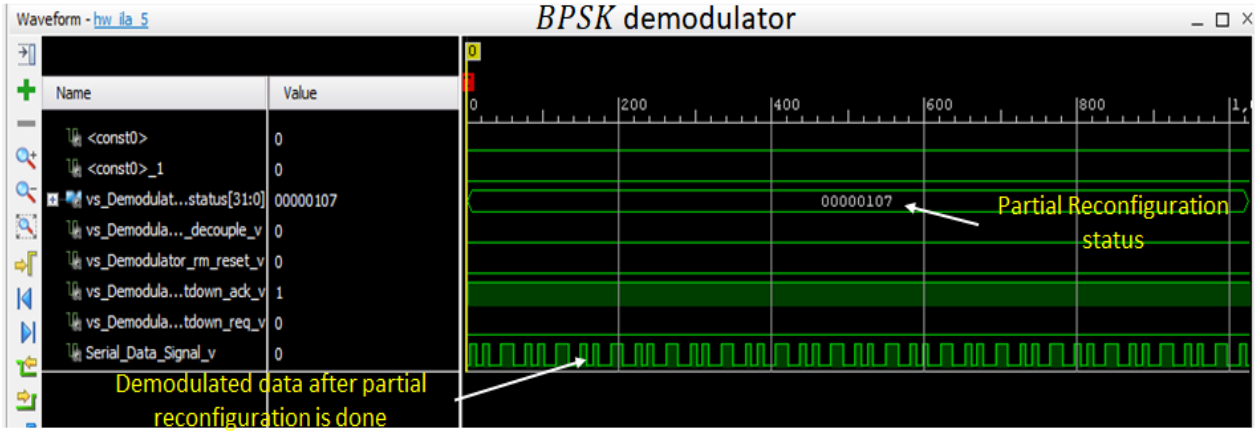


Figure 5-6. Output Results of Partial Reconfiguration Process for RM 1(BPSK).

The value of virtual socket status register of the Virtual Socket Manager (VSM) is equal 00000107 in hexadecimal which equals to "0000_0000_0000_0001_0000_0111" in binary. The description of status register is as following. RM_ID (bits 23:8) = 1, this means RM 1 (BPSK Demodulator) is loaded. SHUTDOWN (bit 7) = 0, this VSM is not in the shutdown state. ERROR (bits 6:3) = 0000, there are no errors. STATE (bits 2:0) = 111, the Virtual Socket is full. Serial data signal is the expected output after performing partial reconfiguration process.

Fig. 5-7 shows the output result according to the other decision value of the classifier. The value of Virtual Socket Status register is equal 00000007. The same status value except RM_ID (bits 23:8) = 0, which means RM 0 (QPSK Demodulator) is loaded.

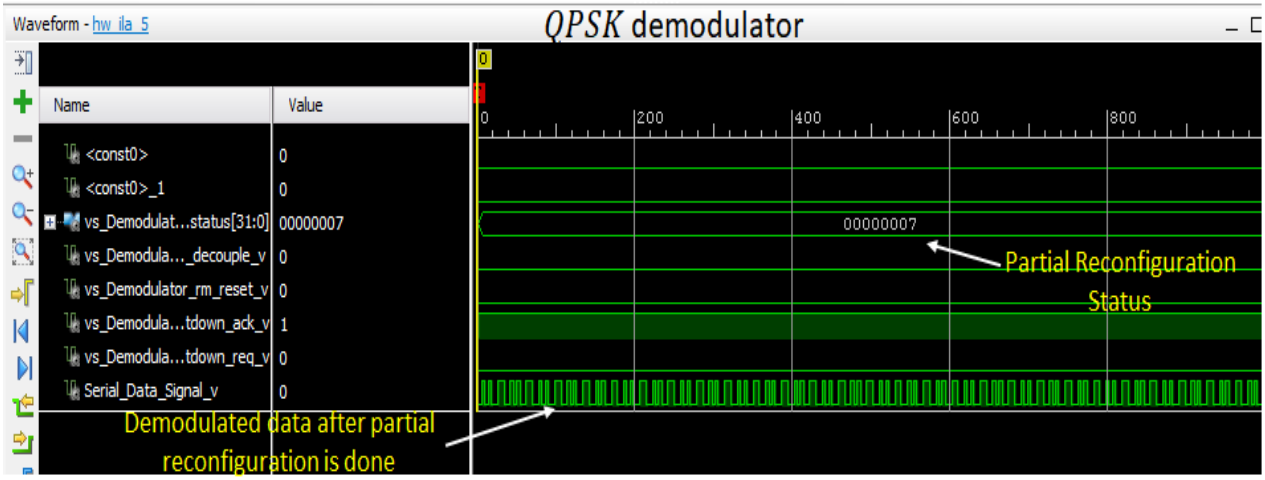


Figure 5-7. Output Results of Partial Reconfiguration Process for RM 0 (QPSK).

Before injection of errors, it is required to determine the essential bits for both static and dynamic sections of FPGA by comparing bits values of EBD with EBC files. 1 in EBD file means the corresponding bit in EBC file is essential and 0 in EBD file means corresponding bit in EBC file is not essential as seen in Fig. 5-8.

According to the determined essential bits, logical address (LA) can be calculated for both static and dynamic sections of the design. A MATLAB program injects errors in LAs of the design randomly (with a rate of error per bit per second) each for the corresponding section. For the dynamic section, error injection is performed before and after partial reconfiguration process.

Using essential bit property provides the user with the percentage of essential bits regarding to all number of configuration bits. In this paper, percentage of Essential Bits (EBs) number of the full design comparing to the total number of bits is about 4%. Percentage of EBs of the dynamic section to EBs of full design is about 0.7% as sees in Fig. 5-9.

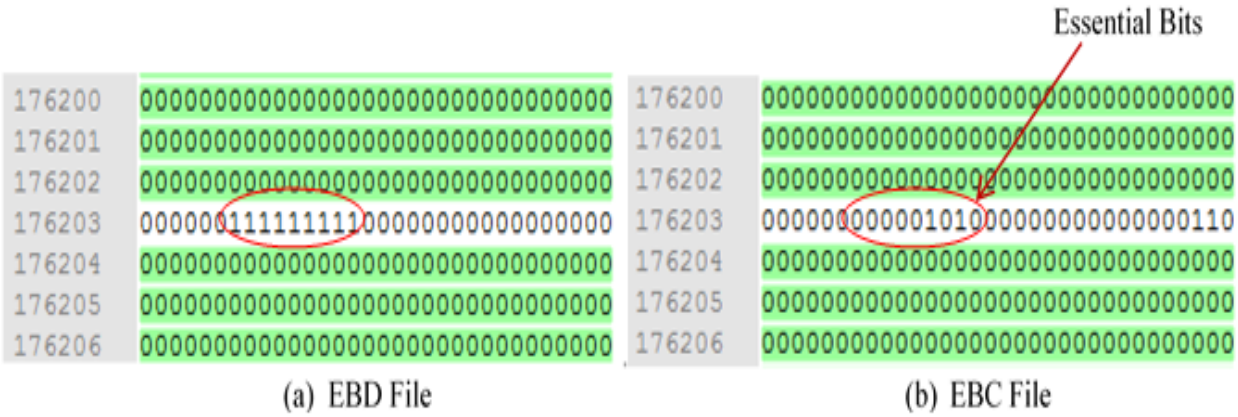


Figure 5-8. Comparing EBD and EBC Files to Determine Essential Bits

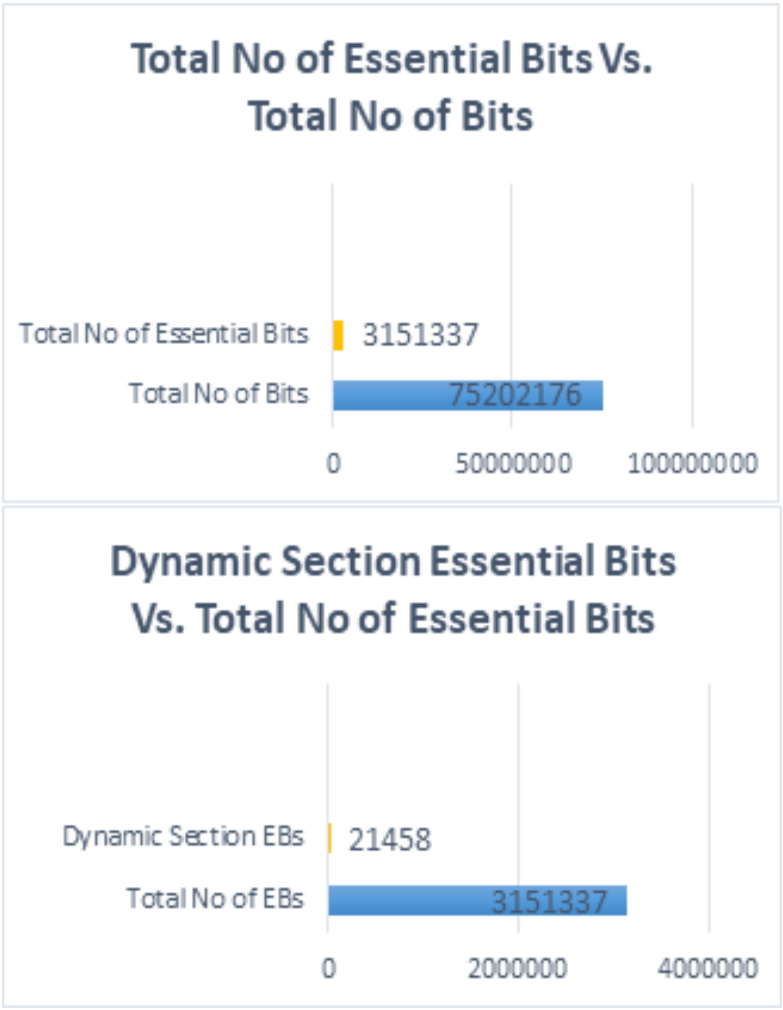


Figure 5-9. Essential Bits Number of Static and Dynamic Sections Comparing to Full Configuration Bits Number

According to Fig. 5-9, the time required for the injection and mitigation of errors in configuration memory using EBs is reduced by a percentage of 96% than the traditional way (injection errors in all configuration memory bits) for the same design. A test is performed by injecting 100 errors in static and dynamic sections, 50 errors in each section, and SEM IP core mitigated these errors successfully.

In Fig. 5-10, a part of the generated file by MATLAB shows the injected errors in essential bits of the configuration memory and the corresponding mitigation results obtained after change the state of the SEM IP core from “Idle” to “Observation” state. The obtained mitigation result for each error is contents of physical and logical addresses, the word and bit number, and the mitigation status.

For example, consider that command N C00001E063, and the binary representation of (1E063) is “00011011000001100011”. According to [84], the first five bits from the right represent the bit number which is “3”, the next 7 bits represent word number which is “3”, and the next 8 bits represent logical address which is “1E”. “COR” indicates that every injected error is determined and corrected successfully.

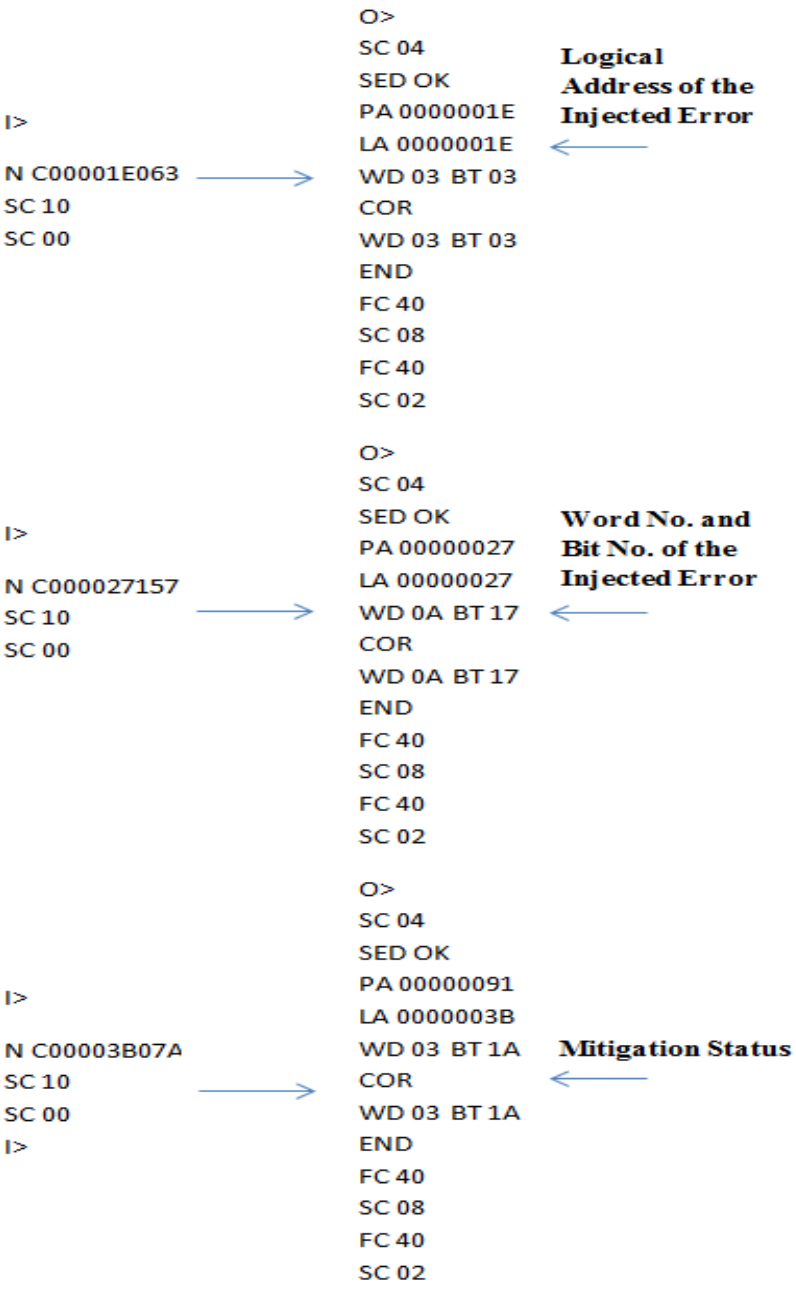


Figure 5-10. A Part of the Generated File of Soft Error Mitigation Process

From the obtained results, Soft error mitigation and partial reconfiguration processes can be performed in DPRDS based on the dual use of ICAP primitive. Sharing ICAP primitive allows SEM and PRC IP cores to access the same ICAP primitive when it is needed for inject and mitigate errors in static section and each RM, and for partial

reconfiguration process respectively. Xilinx essential bits technology has a great effect on the time reduction of error injection process in soft error injection and mitigation process as all. Injection of 100 errors in essential bits for both partial and non-partial regions of DPRDS design is performed. Results shows that the required time for error injection process using essential bits is reduced by 96% compared to the traditional methods that inject errors in the whole configuration memory and SEM IP core mitigates all injected errors successfully.

Chapter 6 : Conclusion and Future Perspective

6.1. Conclusion

The implementation of a DPRDS system can overcome the limited capability of the satellites whereas they can communicate only (demodulate received signal properly) with GCSs that use the same modulation and demodulation schemes in their transceivers. For example, when the satellite uses BPSK modulation/demodulation technique, so it can demodulate the data of the received signal correctly if and only if the modulation of the received signal is BPSK too. Otherwise, the demodulated data will not be correct. The same problem will occur for the communication between a satellite and a relay satellite. The proposed solution of this dissertation discussed the applicability of implementing a DPRDS system in the satellite transceiver to be more generic to permit communication with several relay satellites and GCSs that use different modulation and demodulation techniques without the necessity of using the traditional method required installation of several receivers into satellites so as to reduce the hardware complexity and cost of satellites.

Implementation of a satellite generic receiver using traditional method by installing several receivers into a satellite increases the complexity, power consumption, and cost of a small satellite. A generic receiver promotes the international collaboration and helps the developing and emerging countries to have the access to space activities.

The other objective of a generic receiver is to overcome the communication channel problems by giving the ability to change the modulation techniques based on the communication channel condition and at the same time achieves the required transmission bitrate based on modulation techniques.

The Automatic Digital Modulation Recognition (ADMR) or the classification part of the DPRDS system plays an important role between the detection of the signal and

demodulation stages, which allows a satellite to adaptively respond to the modulation scheme of the received modulated signal correctly after switching the modulation scheme by a GCS/s. There are many applications of ADMR in both civilian and military fields. The ADMR can be used in relay satellites and other satellites where the demodulation and re-modulation of the data are performed onboard. In addition, ADMR can be used to overcome channel problems during the communication session with a satellite as bandwidth congestion/crunch by changing the existing modulation scheme by another more efficient bandwidth modulation scheme.

ADMR can be used to secure the received and transmitted signal by locating, classifying and preventing the unwanted signals. ADMR can be used in interference identification, spectrum monitoring, signal confirmation, electronic warfare, cognitive radio, threat analysis, and signal surveillance.

In this dissertation, the implementation of dynamic partial reconfigurable demodulator system as a prototype of a satellite generic receiver is realized. This prototype based on pattern recognition approach, software defined radio and FPGA dynamic partial reconfiguration technology reduces the hardware complexity and obsolescence, power consumption, and cost of small satellite.

The implemented dynamic partial reconfigurable demodulator system is divided into 2 main sections; automatic digital modulation classification (ADMC), and dynamic partial reconfiguration (DPR) sections. ADMC is responsible of the recognition of the modulation type of the received modulated signal and it consists of features extraction and SVM classifier modules. Features module is to extract the features of the received modulated signal which in this thesis is the discreet wavelet transform (DWT) coefficients level3. The realization of this module on FPGA requires the use 3 DWT stages where each stage is implemented by a finite impulse response filter. Two low pass filters and one high pass filter are required for the implementation. These features are corresponding to the input to the SVM classifier.

SVM classifier is a binary classifier that uses to classify two received modulation types BPSK and QPSK. The classifier has two stages: The training stage which is performed using MATLAB to train the classifier with the features of both BPSK and QPSK modulations. The aim of this stage is to construct the classifier then the resulting parameters which are used to design and implement the classifier on FPGA. The second stage is the classification stage which classifies the received modulation types based on the prior generated parameters (weights, Alpha, bias) during training stage and using radial base function as a SVM kernel function. The classification result is utilized as the input of the DPR section.

The DPR section consists of partial reconfiguration controller (PRC), external memory controller (EMC) and internal configuration access port (ICAP). PRC is the controller of the whole reconfiguration process. It stores the addresses of the BPSK and QPSK demodulator on external flash memory and according to the classification result, which acts as a trigger to DPR controller, and DPR sends the corresponding demodulator address to EMC to fetch the demodulator partial bitstream from external memory then resends this bitstream file to DPR controller. subsequently, DPR controller sends the partial bitstream file to ICAP port to reconfigure the reconfigurable partition by the proper demodulator. After that, the original binary sequence is obtained.

The BPSK and QPSK demodulators are implemented based on SDR using a sine and cosine lookup table instead of sine and cosine wave generator. The demodulation is done by multiplying the samples of the received signal by the sine/cosine samples stored into the lookup tables.

The performance of the system is evaluated by implementing a BPSK and QPSK modulator on one Kintex-7 FPGA board to act as the received signal for dynamic partial demodulator system implemented on another Kintex-7 board to compute the classification rate, classification time, and reconfiguration time.

The system was tested with the presence of AWGN with SNR from -10 to 20 dB. The classification rate for BPSK modulation was 98.7% at -10 dB, increased to 100% at 0 dB,

and kept the same till 20 dB. For QPSK modulation, the classification rate was 1% at -5 dB the increased gradually to 100% at 10 dB and had the same percent till 20 dB.

The classification time was computed as the time between the receiving of the modulated signal till the time the classifier classifies the received signal and generate the classification result or the trigger to the DPR section.

The DPRDS-classifier part of the design has been tested using the received BPSK modulated signal at KIT's GCS which was using to communicate with Horyu-4, and BIRDS projects satellites. Horyu-4 satellite used the BPSK modulation scheme to downlink the payload data. So, I used Horyu-4 satellite received signal to test the classifier part of the DPRDS design. The results state that the classifier succeed to classify the received signal as a BPSK signal but for few times the classifier could not classify the BPSK modulation technique of the received signal. The reasons could be described as the classifier did not trained well on the BPSK modulation in the presence of Doppler shift effect, fading problems and noise of the channel which made the classification of BPSK modulation a little bit difficult for these few times.

The DPRDS-DPR part of the design was implemented by using modular based technique where a design module (demodulator) was replaced with another. Following the procedures provided by Xilinx as described in Chapter 4 helped us to implement this part of the design properly.

To generate the image file that contains full and partial bitstream files, which is stored into an external memory, we have to take care about the type of the external memory and the way the data is stored into memory, where The PRC works in byte addresses because the data is stored in bytes in AXI. Moreover, the memory uses half word addresses because it stores data in half words. So, it is mandatory to compensate the addresses to store every partial bitstream files into the correct address.

The reconfiguration time is computed based on the partial bitstream file and the maximum bandwidth of ICAP port. The size of partial bitstream file is 430,244 bytes and the maximum bandwidth of ICAP is 3.2 Gbps which are resulting in the reconfiguration

file into the reconfigurable partition of the design is about 1.1 msec to make the DPRDS applicable for satellite communication subsystem.

The multiple access of ICAP primitive by several application is proved that the implementation of DPR design will not prevent other application based on the access of the ICAP primitive. In this dissertation, SEM application is used together with the DPRDS design and the results show the possibility of sharing ICAP primitives among different applications.

Using the Xilinx essential bit technology reduces the time required to inject and mitigate the errors in the DPRDS design by a 96%. The main reason is essential bit technology based of injecting the error only into the bits of the configuration frame that are used by the design and ignoring the other bits. To get the essential bits, EBC and EBD have to be generated for comparing the bits existing into EBD file with the corresponding bits of EBC file, where bits of value 1 in EBD file mean the corresponding bits in EBC file are essential, otherwise, the corresponding bits in EBC file are not essential.

According to the resulting essential bits, a MATLAB script is used to calculate the LA of these bits for injecting and mitigating the errors, through UART interface, in both static and dynamic sections of the DPRDS design on the conditions of SEM IP core mentioned in chapter 5.

6.2. Future Perspective

In this research, we described a DPRDS that can be used as a prototype of a generic receiver. Regarding to the high performance of the system as appeared from the obtained results, this system can be used to satellite communication systems. I hope to apply the system on a real satellite to test its performance in runtime in space to show the performance and reliability of this system.

The DPRDS has to be developed in satellite communication system. The DPRDS has to recognize as much as possible of modulation techniques that use in satellite telecommunication systems. DPRDS will be developed to recognize AFSK, FM, MPSK,

MSK, and GMSK modulation techniques which are mostly used for satellite telecommunication systems.

Enhancing the performance and capabilities of the DPRDS by increasing the recognized modulation techniques resulting in the enhancement and development of the classifier as well. The SVM classifier used in this research is a binary classifier, which means it can classify only two modulation techniques. So SVM classifier needs to be developed to realize the classification of several modulation techniques.

There are two techniques are using to transform SVM classifier from binary to multi-classifier. One-Against-One (OAO) and One-Against-All (OAA) techniques. In OAO, N-SVMs (where N is the number of classes too) are constructed and each SVM is trained to separate one class from another class. The classification decision is based on the voting result of these SVMs. In OAA, N-SVMs (where N is the number of classes too) are constructed and each SVM is trained to separate one class from another class. Only the SVM trained on an input class has a positive response.

For more enhancements, several features extraction methods and the combination of some of them have to be applied to obtain a perfect feature extraction mechanism. Extraction of the features from different properties of the signal as spectrum, frequency, and phase have to be considered.

I hope applying this system in one of the satellites that are developing in ESP to stand on the performance and reliability of the system and disseminate the results to the world via academic research journals.

List of Publications

Journals:

1- Mohamed Elhady Keshk, Dagvasumberel Amartuvshin, Rafael Rodriguez, Kenichi Asami, “Design SVM Classifier for Automatic Modulation Recognition System”, UNISEC Space Takumi Journal, Vol. 7, No. 1, pp. 1-14, July 2018.

2- Mohamed Elhady Keshk, Kenichi Asami, “Fault Injection in dynamic partial reconfiguration design based on essential bits”, Journal of aeronautics and space technologies (JAST), Vol. 11, No. 2, pp. 25-33, July 2018.

Conferences:

1- Mohamed Elhady Keshk, Kenichi, “Dynamic Partial Reconfigurable Demodulators in Automatic Modulation Recognition Systems for satellite receivers”, AIAA Space and Astronautics Forum and Exposition, 12 - 14 September 2017, Orlando, FL, USA.

2- Mohamed Elhady Keshk, Kenichi Asami, “FPGA-based Automatic Modulation Recognition System for Small Satellite Communication Systems”, Proc. Of 31st Annual AIAA/USU Conference on Small Satellites, August 2017, Utah, USA.

3- Mohamed El-Hady M. Keshk, Kenichi Asami, “Comparison between Generation of Analog and Digital QPSK Modulation for Satellites Communication Systems”, Proc. of 67th International Astronautical Congress IAC16, September 2016, Mexico.

4- Mohamed Elhady Keshk, Kenichi Asami, “Implementing QPSK Modulator with Low Resources Utilization and Power Consumption for Satellites Communication Systems using Kintex-7”, Proc. Of International Workshop on Satellite & Space Missions: Developments and Applications, June 2016, Egypt.

References

[1] K. Pedersen, “Thoughts on International Space Cooperation and Interests in the Post-cold War World”, *Space Policy Journal*, Vol. 8, Issue 3, pp. 205-220, Aug., 1992.

[2] E. Sadeh, J. Lester, W. Sadeh, “Modeling International Cooperation for Space Exploration”, *Space Policy Journal*, Vol. 12, Issue 3, pp. 207-223, Aug. 1996.

[3] B. Gupta, K. Raju, “Space Exploration by India and Socio-economic Cooperation with SAARC Countries”, *A Journal of International Affairs*, Vol. 72, Issue 3, pp. 278-289, Oct., 2016.

[4] S. Ciccarelli, “Space Cooperation with Developing Countries: The Case of Morocco”, *Journal of Middle Eastern Geopolitics*, pp. 129-134, 2006.

[5] C. Gomez, “The Difficulties of Space Cooperation in Latin America: Perspectives and Solutions”, *Proc. of the International Astronautical Congress*, 2016.

[6] R. Jakhu, “The Case for Enhanced India–Canada Space Cooperation”, *Space Policy Journal,* Vol.25, Issue 1, pp. 9-19, Feb., 2009.

[7] W. Huntley, “Canada-China Space Engagement: Opportunities and Prospects”, China Paper series 20, Centre of International Relations, University of British Colombia, Canadian International Council, Feb., 2011.

[8] R. Farrell, M. Sanchez, and G. Corley, “Software-Defined Radio Demonstrators: An Example and Future Trends”, *International Journal of Digital Multimedia Broadcasting*, Vol. 2009, pp. 1-12, Jan., 2009.

[9] W. Lie, W. Feng-yan, “Dynamic Partial Reconfiguration in FPGAs”, Third *International Symposium on Intelligent Information Technology Application*, pp. 445-448, 2009.

[10] D. Raychaudhuri and N. Mandayam, “Frontiers of Wireless and Mobile Communications”, *Proceedings of the IEEE*, Vol.100, No.4, April, 2012.

[11] O. Press, “The Image Empire: A History of Broadcasting in the United States from 1953”, *Oxford University Press*, Vol. III, Nov., 1970.

[12] L. Kleinrock, “An Early History of the Internet [History of Communications]”, *IEEE communications Magazine*, Vol. 48, Issue 8, Aug., 2010.

[13] N. Abramson, “Development of the ALOHANET”, *IEEE Transactions on Information Theory*, Vol. 31, Issue 2, Mar., 1985.

[14] U. F. C. Commission, “Authorization of Spread Spectrum Systems”, *under Parts 15 and 90 of the FCC Rules and Regulations”, General Docket*, pp. 81-413, 1985.

[15] C. Rinaldi, “Performance Evaluation and Optimization of an OMAP Platform for Embedded SDR Systems”, *Master of Science Thesis*, School of Information and Communication Technology, Royal Institute of Technology (KTH), Stockholm, Sweden, Feb., 2011.

[16] R. Hosking, “Software Defined Radio Handbook”, *Handbook*, 10th edition, Apr., 2013.

[17] A. Lalge, M. Karpe, S. Bhandari, “Software Defined Radio Principles and Platforms”, *International Journal of Advanced Computer Research*, Vol. 3, Number-3, Issue 11, Sep., 2013.

[18] R. Farrell, M. Sanchez, and G. Corley, “Software-Defined Radio Demonstrators: An Example and Future Trends”, *International Journal of Digital Multimedia Broadcasting*, Vol. 2009, pp. 1-12, Jan., 2009.

[19] X. Feng and J. Li; “A Hierarchical Digital Modulation Classification Algorithm for Adaptive Wireless Communication Systems”, *Wireless Personal Communications*, Vol. 39, pp. 321-326, 2006.

[20] D. Grimaldi, S. Rapuano, and L. DeVito, “An Automatic Digital Modulation Classifier for Measurement on Telecommunication Networks”, *IEEE Trans on Instrumentation and Measurements*, Vol. 56, 2007.

[21] T. Welch, T. Kent, C. Wright, and M. Morrow, “An Affordable Software Defined Radio”, *Digital Signal Processing and 5th IEEE Signal Processing Education Workshop*, pp. 791-796, Marco Island, FL, (2009).

[22] M. Valipour, M. Homayounpour and M. Mehralian, “Automatic Digital Modulation Recognition in Presence of Noise Using SVM and PSO”, *6th International Symposium on Telecommunications (IST)*, 2012.

[23] O. Dobre, A. Abdi, Y. Bar-Ness and W. Su, “A Survey of Automatic Modulation Classification Techniques: Classical Approaches and New Trends”, *part of this work was published in Sarnoff Symposium*, Princeton, NJ, USA, Apr., 2005.

[24] U. Satija, M. Mohanty, B. Ramkumar, “Automatic Modulation Classification using S- transform based Features”, *2nd International, Conference on Signal Processing and Integrated Networks (SPIN)*, pp.708-712, 2015.

[25] R. Li, “Modulation Classification and Parameter Estimation in Wireless Networks”, *Master thesis*, Department of Electrical Engineering, Syracuse University, June, 2012.

[26] O. Mohamed, “FPGA Recognition of Digital Modulation with Adaptive Receiver”, *Doctoral thesis*, Computers and Systems Department, Faculty of Engineering, Zagazig University, 2009.

[27] B. Blodget, C. Bobda, M. Huebner, and A. Niyonkuru, “Partial and Dynamically Reconfiguration of Xilinx Virtex-II FPGAs”, *International Conference on Field Programmable Logic and Applications*, Vol. 3203 of Lecture Notes in Computer Science, pp. 801–810, 2004.

[28] Xilinx, “7 Series FPGAs Configuration Logic Blocks”, UG474, V1.8 Sep., 2016.

[29] P. Kalra, A. Kukde, and B Venkataramani, “CORDIC based BPSK Modulator”, *5th International Conference on Computer and Communication Technology*, pp. 335-339, 2014.

[30] Y. Tang, X. Lv, “Research on the Modulation and Demodulation of BPSK and BDPSK Simulator Based on Matlab”, *International Conference on Electrical and Control Engineering, IEEE,* pp. 1239-1241, 2011.

[31] R. Das, “FPGA Implementation of Digital Modulation Schemes BPSK and QPSK Using VHDL”, *International Journal of Electronics and Communication Technology*, Vol. 5, pp. 111-113, 2014.

[32] S. Popescu, A. Gontean, and D. Ianchis, “QPSK Modulator on FPGA”, SISY 2011, *IEEE 9th International Symposium on Intelligent Systems and Informatics*, pp. 359-364, Sep. 8-10, 2011.

[33] B. Umesharaddy, B. Sujatha, “Optimization of QPSK MODEM with AWGN Implemented in FPGA”, *International Conference on Inventive Systems and Control*, pp. 1-6, 2017.

[34] U. Satija, M. Mohanty, B. Ramkumar, “Automatic Modulation Classification using S-transform based Features”*, 2nd International Conference on Signal Processing and Integrated Networks*, pp.708-712, 2015.

[35] D. Digdarsini, M. Kumar, G. Khot, T. Ram, V. Tank, “FPGA Implementation of Automatic Modulation Recognition System for Advanced SATCOM System”, *International Conference on Signal Processing and Integrated Networks*, pp. 464-469, 2014.

[36] A. Elrharras, R. Saadane, M. Wahbi, and A. Hamdoun, “Signal Detection and Automatic Modulation Classification Based Spectrum Sensing Using PCA-ANN with Real Word Signals”, *Applied Mathematical Sciences*, Vol. 8, pp. 7959 – 7977, 2014.

[37] Y. Ge., D. Daut, K. Ho, Canute V., “Automatic Modulation Recognition Using DWT-Based Signal Templates”, *Proceedings of SDR-WInnComm*, pp.329-337, 2013.

[38] M. Farhang, H. Bahramgiri, H. Dehghani, “Automatic Modulation Classification Using an 8PSK Demodulator for Variants of QPSK”, *6’Th International Symposium on Telecommunications*, pp. 317-321, 2012.

[39] M. Farhang, H. Bahramgiri, H. Dehghani, "Multi-Receiver Modulation Classification for Satellite Communications Signals", *IEEE International Conference on Signal and Image Processing Applications (ICSIPA2011),* pp. 569-573, 2011.

[40] J. Grajal, O. Yeste-Ojeda, M. Sanchez, M. Garrido and M. Lopez-Vallejo, "Real Time FPGA Implementation of an Automatic Modulation Classifier for Electronic Warfare Applications", *19th European Signal Processing Conference*, pp. 1514-1518, 2011.

[41] M. Aslam, Z. Zhu, and A. Nandi, "Automatic Digital Modulation Classification Using Genetic Programming with K-Nearest Neighbor", *Proc. Of The 2010 Military Communications Conference - Unclassified Program - Waveforms and Signal Processing Track*, pp. 512-517, 2010.

[42] R. Graczyk, M. Stolarski, M. Palau, "Dynamic partial FPGA reconfiguration in space applications," *Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments*, Vol. 8454, pp. 1-7, 2012.

[43] S. Bhandari, S. Subbaraman, S. Pujari, "Digital Signal Modulator on FPGA using On the Fly Partial Reconfiguration", *Proc. Of International Conference on Advances in Computing, Control, and Telecommunication Technologies*, pp. 711-713, 2009.

[44] C. Vennila, K. Suresh, R. Rathor, G. Lakshminarayanan, S. Ko, "Dynamic Partial Reconfigurable Adaptive Transceiver for OFDM Based Cognitive Radio", *26th IEEE Canadian Conference Of Electrical And Computer Engineering (CCECE)*, pp. 1-4, 2013.

[45] O. Hiari, and R. Mesleh "A Reconfigurable SDR Transmitter Platform Architecture for Space Modulation MIMO Techniques", *IEEE Access*, Vol. 5, pp. 24214 – 24228, Oct., 2017.

[46] S. Shrestha and K. Wahid, "Hybrid DWT-DCT Algorithm for Biomedical Image and Video Compression Applications", *Proc. of the 10th IEEE International Conference on Information Sciences, Signal Processing and their Applications*, pp. 280-283, 2010.

[47] R. Jiang, D. Crookes, "FPGA implementation of 3D Discrete Wavelet Transform for Real-time Medical Imaging", *Proc. of 18th European Conference on Circuit Theory and Design,* pp. , 2007.

[48] R. Sevakula, R. Kumar, and N. Verma, "Support Vector Machine for Large Databases as Classifier", *Swarm, Evolutionary, and Memetic Computing. Springer Berlin Heidelberg*, pp. 303-313, 2012.

[49] H. Chen, B. Yang, S. Wang, G Wang, D. Liu, H. Li, W. Liu, "Towards an Optimal Support Vector Machine Classifier using a Parallel Particle Swarm Optimization Strategy," *Applied Mathematics and Computation*, Vol. 239, pp. 180–197, 2014.

[50] K. Parikh, T. Shah, “Support Vector Machine – a Large Margin Classifier to Diagnose Skin Illnesses”, *Proc. of 3rd International Conference on Innovations in Automation and Mechatronics Engineering (ICIAME)*, pp. 369-375, 2016.

[51] C. Chao and M. Horng, “The Construction of Support Vector Machine Classifier Using the Firefly Algorithm”, *Computational Intelligence and Neuroscience*, Vol. 2015, pp. 1-8, 2015.

[52] S. Aurangabadkar and M. Potey, “Support Vector Machine Based Classification System for Classification of Sport Articles”, *Proc. of International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, pp. 146-150, 2014.

[53] M. Rahman, M. Kim, J. Takada, "Emergency Radio Identification by Supervised Learning based Automatic Modulation Recognition", *Asia Pacific Radio Science Conference (AP-RASC)*, pp. 1-10, Sep., 2010.

[54] G. Ghiwala , P. Thaker , G. Amin, “Realization of FPGA based numerically Controlled Oscillator”, *IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)*, Vol. 1, Issue 5, (Jan. – Feb.), PP 07-11, 2013.

[55] G. Ao “The Basic Principle and FPGA Implementation of NCO”, *Second International Conference on Instrumentation & Measurement, Computer, Communication and Control*, pp. 90-94, 2012.

[56] T. Zarey, J. Aboueiyy, K. Plataniotisyy, “KGDA for Signal Classification in Cognitive Radio”, *7th International Symposium on Telecommunications (IST'2014)*, pp. 1106-1112, 2014.

[57] M. Petrova, P. Mahonen and A. Osuna, “Multi-Class Classification of Analog and Digital Signals in Cognitive Radios using Support Vector Machines”, *International Symposium on Wireless Communication Systems (ISWCS)*, pp. 986-990, 2010.

[58] A. Ebrahimzadeh and G. Ardeshir, “A New Signal Type Classifier for Fading Environments”, *Journal of Computing and Information Technology - CIT 15*, issue 3, pp. 257–266, 2007.

[59] H. Hu, J. Song and Y. Wang “Signal Classification Based on Spectral Correlation Analysis and SVM in Cognitive Radio”, *Proc. of 22nd International Conference on Advanced Information Networking and Applications*, pp. 383-387, 2008.

[60] M. Steiner, “Spectrum Sensing and Blind Automatic Modulation Classification in Real Time”, *Master thesis*, Electrical and Computer Engineering, faculty of the Virginia Polytechnic Institute and State University, 2011.

[61] Z. Xing, J. Zhou, “Automatic Modulation Recognition of PSK Signals Using Nonuniform Compressive Samples Based on High Order Statistics”, *Proc. of IEEE*

International Conference on Communication Problem-solving (ICCP), pp. 611-614, 2014.

[62] K. Bhuvaneswari,V. SrinivasaRao, "Dynamic Partial Reconfiguration in Low-Cost FPGAs", *International Journal of Scientific & Engineering Research*, Vol. 4, Issue 9, pp. 1410-1413, Sep., 2013.

[63] Xilinx, "Partial Reconfiguration User Guide", UG702, Ver.13.1, March, 2011.

[64] Xilinx, "Vivado Design Suite User Guide, Partial Reconfiguration", UG909, Ver. 2016.1, Apr., 2016.

[65] T. Sasamal, R. Prasad, "Module Based And Difference Based Implementation of Partial Reconfiguration On FPGA: A Review", *International Journal of Engineering Research and Applications (IJERA)*, Vol. 1, Issue 4, pp. 1898-1903, 2011.

[66] S. Kota, A. Gupta, S. Nayak, and S. Varma, "Module Based Implementation of Partial Reconfiguration Using VHDL on Xilinx FPGA", *International Journal of Recent Trends in Engineering*, Vol. 2, No. 7, pp. 19-21, Nov., 2009.

[67] M. Feilen, "Dynamic Partial Self-Reconfiguration of FPGAs for Digital Broadcasting Receiver Systems", *Ph.D. dissertation*, Faculty of Electrical Engineering and Information Technology, the Technical University of Munich, Feb., 2017.

[68] Xilinx, "Vivado Design Suite User Guide, Partial Reconfiguration", UG909, Ver. 2017.1, Apr., 2017.

[69] Xilinx, "Partial Reconfiguration Controller v1.0, LogiCORE IP Product Guide", PG193, April, 2016.

[70] Xilinx,"Vivado Design Suite Tutorial Partial Reconfiguration", UG947, Ver. 2016.1, April, 2016.

[71] T. Nidhin, A. Bhattacharyya, R. Behera, T. Jayanthi, K. Velusamy, "Understanding Radiation Effects in SRAM-based Field Programmable Gate Arrays for Implementing Instrumentation and Control Systems of Nuclear Power Plants", *Nuclear Engineering and Technology Journal*, pp. 1589-1599, 2017.

[72] F. Brosser E. Milh, "SEU Mitigation Techniques for Advanced Reprogrammable FPGA in Space", *Master dissertation*, Dept. Computer Science and Eng., chalmers Univ. of Technology, Gothenburg, Sweden, 2014.

[73] M. Ibrahim, K. Asami and M. Cho, "LEO Single Event Upset Emulator for Validation of FPGA Based Avionics Systems", *Trans. JSASS Aerospace Tech*., Vol. 12, No. ists29, pp. Tf_19-Tf_25, 2014.

[74] B. Pratt, M. Caffrey, J. Carroll, P. Graham, K. Morgan, and M. Wirthlin, "Fine-Grain SEU Mitigation for FPGAs Using Partial TMR", *IEEE Trans. on Nuclear Science*, Vol. 55, NO. 4, pp.2274-2280, Aug., 2008.

[75] R. Gosheblagh and K. Mohammadi, “Dynamic Partial based Single Event Upset (SEU) Injection Platform on FPGA”, *International Journal of Computer Applications,* Vol. 76, No.3, pp. 19-24 Aug. 2013.

[76] H. Quinn and M. Wirthlin, “Validation Techniques for Fault Emulation of SRAM-based FPGAs”, *IEEE Transaction on Nuclear Science*, Vol. 62, NO. 4, pp. 1487-1499, Aug., 2015.

[77] U. Legat, A. Biasizzo, and F. Novak, “SEU Recovery Mechanism for SRAM-Based FPGAs”, *IEEE Transaction on Nuclear Science*, Vol. 59, NO. 5, pp. 2562-2571, OCT., 2012.

[78] G. Asadi, M. Tahoori, “Soft error mitigation for SRAM-based FPGAs”, *Proc. of 23rd IEEE VLSI Test Symposium*, 2005.

[79] A. Sari, M. Psarakis, “Scrubbing-based SEU Mitigation Approach for Systems-on-Programmable-Chips”, *International Conference on Field-Programmable Technology*, pp. 1-8, Dec., 2011.

[80] F. Abate, L. Sterpone, C. Lisboa, L. Carro, and M. Violante, “New Techniques for Improving the Performance of the Lockstep Architecture for SEEs Mitigation in FPGA Embedded Processors”, *IEEE Trans. on Nuclear Science*, Vol. 56, NO. 4, pp. 1992-2000, Aug., 2009.

[81] C. Carmichael and C. Tseng, “Correcting Single-Event Upsets in Virtex-4 FPGA Configuration Memory”, *Xilinx*, XAPP1088 Ver. 1.0, Oct., 2009.

[82] R. Velazco, G. Foucard, and P. Peronnard, “Combining Results of Accelerated Radiation Tests and Fault Injections to Predict the Error Rate of an Application Implemented in SRAM-Based FPGAs”, *IEEE Trans. on Nuclear Science*, Vol. 57, NO. 6, pp. 3500-3505, Dec., 2010.

[83] H. Ziade, R. Ayoubi, R. Velazco, and T. Idriss, “A New Fault Injection Approach to Study the Impact of Bit flips in the Configuration of SRAM-Based FPGAs”, *The International Arab Journal of Information Technology*, Vol. 8, No. 2, pp. 155-162, April 2011.

[84] Xilinx, “Soft Error Mitigation Controller v4.1”, *Xilinx*, PG036, April, 2017.

[85] SPENVIS online package website: www.spenvis.oma.be.

[86] J. Ayer, “Dual Use of ICAP with SEM Controller”, *Xilinx*, XAPP517, Ver. 1.0, Dec., 2011.

[87] R. Le, “Soft Error Mitigation Using Prioritized Essential Bits”, *Xilinx*, XAPP538 Ver. 1.0, April, 2012.

[88] A. Smith, M. Evans, and J. Downey, “Modulation Classification of Satellite Communication Signals Using Cumulants and Neural Networks”, *Proc. of Cognitive Communications for Aerospace Applications Workshop*, pp. 1-8, Jun. 2017.