Design and Realization of Novel Adaptive Digital Beam Former Architecture for Active Phased Array Radar

Submitted in partial fulfilment of the requirements

for the award of the degree of

Doctor of Philosophy

by

Govind Rao Doddamani
(Roll No. 701144)

Supervisor (s):

Dr. T. Kishore Kumar

Dr. A. Vengadarajan

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY

WARANGAL – 506004, INDIA.

February 2019

APPROVAL SHEET

This Thesis entitled **"Design and Realization of Novel Adaptive Digital Beam Former Architecture for Active Phased Array Radar"** by **Mr. Govind Rao Doddamani** is approved for the degree of **Doctor Philosophy.**

Examiners

…………………………….

……………………………..

Supervisor (s)

……………………………..

Dr. T. Kishore Kumar

ECE Dept., NIT Warangal

……………………………..

Dr. A. Vengadarajan, Sc-G

LRDE, DRDO, Bangalore

Chairman

……………………………..

ECE Dept., NIT WARANGAL

Date:

DECLARATION

This is to certify that the work presented in the thesis entitled “**Design and Realization of Novel Adaptive Digital Beam Former Architecture for Active Phased Array Radar**” is a bonafide work done by me under the supervision of Dr. T. Kishore Kumar, ECE Dept., NIT Warangal and Dr. A. Vengadarajan, Director, DG ECS Office, DRDO, Bangalore, India and was not submitted elsewhere for the award of any degree.

I declare that this written submission brings my own ideas in my own words and others ideas or words have not been included. I have sufficiently quoted and referenced the original sources wherever it was used. I have followed to all principles of academic honesty and integrity and have not misused or fabricated or falsified any idea/date/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been suitably cited or from whom proper permission has not been taken when needed.

(D. Govind Rao)

(Roll No. 701144)

Date:

National Institute of Technology, Warangal

(Deemed University)

CERTIFICATE

This is to certify that the thesis entitled “**Design and Realization of Novel Adaptive Digital Beam Former Architecture for Active Phased Array Radar**” being submitted by **Mr. Govind Rao Doddamani** in partial fulfilment for the award of the degree of **Doctor of Philosophy** to the Department of Electronics and Communication Engineering of National Institute of Technology Warangal, is a record of bonafide research work carried out by him under our supervision and has not been submitted elsewhere for any degree.

Dr. T. Kishore Kumar,
(Supervisor)
Dept. of Electronics & Communication Engineering,
National Institute of Technology,
Warangal – 506004, India.

Dr. A. Vengadarajan,
(Co-Supervisor)
LRDE,
DRDO,Govt of India
C. V. Raman Nagar
Bangalore – 560093, India.

ACKNOWLEDGEMENTS

At the outset, I take immense pleasure to convey my sincere gratitude to my supervisors **Dr. T. Kishore Kumar, Prof. N.S. Murthy and Dr. A. Vengadarajan**, Director D.G ECS Office, DRDO, Bangalore, for their perpetual encouragement and supervision. Their steady influence throughout my Ph.D. career has oriented me in a proper direction and supported me with promptness and care. They listened to my ideas and discussions led frequently to key insights and gave full support even in the state of despair. I truly appreciate their logical and thought provoking advice both technically and morally which I will follow for the rest of my life.

I thank all the faculty and non-teaching staff of Dept. of ECE at NIT Warangal who helped me during the course. I am also grateful to Prof. N.V.S.N. Sarma, Dr. B. Laxmi and Dr. L. Anjeneyulu Department of Electronics & Communication Engineering, for his invaluable assistance and suggestions that he shared during my research tenure.

I take this privilege to thank all my Doctoral Scrutiny Committee members, Prof. K.S.R. Krishna Prasad, Professor of Department of Electronics & Communication Engineering, Prof. Vinod Kumar, Electrical & Electronics Engineering Department, Prof. T.Ramesh, Computer Science Engineering Department and Prof. G. Radhakrishnamacharya, Mathematics Department, for their detailed analysis, productive suggestions and exceptional advice during the progress of this research work.

I thank my nation India and all the people supported me directly and indirectly including Director LRDE, DRDO Bangalore for supporting me to carry out my research work at the NIT Warangal.

I also extend my heartfelt appreciation to all my colleague scholars, friends and well-wishers who helped to write my thesis with their support. Finally I would like to acknowledge my biggest debt to my parents and my wife for their constant support.

(D. Govind Rao)

ABSTRACT

The concept of beam forming in phased array was implemented with analog or mechanical solutions. The present day need is to have every element level beam formation instead of a group of elements combined in the RF domain. Element level beam formation was a major challenge to realize in hardware. Due to this limitation researchers have developed sub array level beam formers for phased array application but this brings many limitations like roll off of the sub array pattern which causes a gain loss in the re-steered direction (Reduces the range of the radar) and produces grating lobes.

The digital systems have become powerful to carry out the huge number of tasks required for real-time digital beam formation. Today many applications are using beam forming to enhance the effective channel utilization both in frequency and space, more digital dedicated architectures are proposed for the parallel and pipelined processing for communication applications. With the help of reconfigurable system, the same hardware platform can be reutilised for different applications with different processing needs. This thesis designed a novel method of hardware design and realization of adaptive digital beam former for phased array Radar application.

This research work was carried out considering a typical case of sixteen element planar phased array. Initially multiple digital beam formation is achieved using the computation of weights off line and stored in the memory of the digital board and 4 and 8 receive beams were formed. The digital beam formation with fixed weights architecture is designed and functional simulation has been carried out. The same architecture is implemented on the Virtex-VI FPGA based hardware and multiple receive beams were formed.

Further the architecture development is extended to computation of adaptive weights online and a parallel pipelined architecture is designed. A survey of various algorithms has been carried out for adaptive weight calculation and QR-Decomposition based Recursive Least Square (QRD-RLS) is identified as most suitable. Since this algorithm is computationally complex and takes more time for the optimal weight calculation a modified algorithm is developed for online weight calculation. The Inverse QRD-RLS algorithm is a most efficient adaptive algorithm and from the hardware realisation point of view it is optimised. A systolic structure method can be employed to calculate the weights in a given time and is numerically stable compared to other traditional algorithms.

The most suitable algorithm for pipelined and parallel implementation architecture is the inverse QRD-RLS algorithm. For phased array radar applications optimal weights must be computed in a short time of the order of few micro seconds with very good accuracy. In such cases Inverse QRD-RLS is most suitable.

The novel architecture is developed for sixteen element planar phased array to form multiple receive beams for radar applications. Present day FPGA's are capable of concurrent processing, a three FPGA architecture is developed to form the adaptive beams simultaneously.

The outcome of this research work is realisation of generic, modular, scalable adaptive digital beam former architecture for a sixteen element planar antenna array configuration. This also can be extended for larger dimension in the phased array application.

Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

RLS	Recursive Least Squares
QRD	Q-R Decomposition
CRLS	Constrained Recursive Least Squares
LMS	Least Mean Square
FPGA	Field Programmable Gate Array
DDC	Digital Down Conversion
ADC	Analog to Digital Converter
NCO	Numerically Controlled Oscillator
DDS	Direct Digital Synthesizers
T/R Module	Transmit-Receive Module
LUT	Look-Up Table
IQRD	Inverse QR Decomposition
ADBF	Adaptive Digital Beam Forming

Chapter 1

Introduction

Chapter 1.

Introduction

1.1 Beam Formation

Phased array antennas have played a crucial role in development of multifunction radars. There are numerous benefits of electronic scanning for multiple targets, search and track functions due to sufficient beam agility. The next generation of phased array antennas will offer improvements in radar functionality utilizing Digital Beam Forming (DBF) in place of its analog counterpart [112]. However, DBF requires higher computational throughput and more receivers to digitally perform the same operations that were formerly done with analog hardware. But due to advancement in the technology of digital computing it has become increasingly easier to accommodate the more demanding computing requirements.

A conventional analog phased array antenna sums the signals from the individual antenna elements through radio frequency combiner networks made up of one or more stages of phase shifter, attenuators, amplifiers and time delay networks. After the beams are formed in analog receivers, an ADC will be used to produce digital signals that are input to the radar signal and data processing computers. In a radar system with DBF architecture the receivers are placed prior to the final stage of combining so that some or all of the beam forming occurs in a digital computer. From an array of elements in the phased array configuration every element will have a low noise amplifier and then go directly into a receiver where the analog is converted to digital in phase and quadrature samples. The digital I and Q samples are then sent to a digital beam forming processor where the beam forming is done in a high speed computer.

A number of array signal processing techniques can be illustrated using the DBF architecture, including the following that can enhance the capability of radar systems,

1. For improved search occupancy by digital steering of beams on receive mode

2. Electromagnetic interference (EMI) mitigation and adaptive cancellation for jammer

3. Improved metric accuracy by high-resolution angle estimation of targets and jammers.

Also a significant amount of error due to costly analog time-delay units can be reduced by replacing them with infinite precision digital time delay wideband DBF array designs. As compared to a conventional analog array, where entire sum or monopulse difference beam is lost due to failures, a DBF array is more robust to receiver failures since only the part of the array feeding that receiver is lost.

Digital beam former design is an essential part of radar and information exchange architectures. There are numerous ways that beam forming is currently utilized [3][7]. The optimality of the system used in signal reception decreases due to sensitivity in SNR caused by unwanted signal which enters the system neither by the main lobe nor by the side lobes of the patterned beam. The processing of the signal is done by the signal processors for the signals which are received and then the beam is formed in the directions as specified [24][27].

Presently, efficient beam forming is employed in the filter algorithms which are adaptive and it can be utilized in the array system that comprises of the radar such that the intended signal is preserved in the occurrence of noise signal [25]. Due to the increase in radar traffic the interference suppression becomes more essential in those systems where adaptive beam forming has its major usage. The signals and interference properties are distinguished in the active phased planar array through beam forming where it comprises an array of sensors that are independent so that it gives the samples of signal which are received in space [22].

Hereby outputs from the sensor are modeled by a filter that can be transverse to produce output in the form of the beam. The basic aim of the modifying filter is to defend the target signal when the noise signal is cancelled. Moreover the modifying filter provides capacity to the system that consists of an array of antenna such that it can automatically capture the occurrence of the intruding signals to precisely suppress the noise signals which simultaneously enhance the referenced signal the QRD RLS algorithms are opted to be a better approach for those applications where convergence speed is paramount [27].

1.2 Phased Arrays

In a phased array antenna, the effective radiation pattern of the array is reinforced/suppressed in a desired/undesired direction by varying the relative phases of the respective signals feeding the antennas.

Multiple active antennas are grouped together and coupled to a common source to result into a directive radiation pattern in an antenna array. Normally the directivity of the antenna array is related to the spatial relationship of the individual antennas [5]. An "Active antennas" is one in which the signal input controls the source of energy. A multiband television antenna is one example of this nature [4].

The desired radiation pattern is achieved by changing the phase of the various elements. The limits are determined by:

(a) The array configuration.
(b) The array size.
(c) The elements radiation pattern.

Phased Array main benefits are [32]

- A large structure need not be rotated to cover the entire space.
- The steering can be done in electronic mode hence faster steering in the desired direction.
- Solid-state transmitters are used in the entire phased array instead of a single RF sources. Hence warm-up time of the array is less, complex RF feed system not required, and single-point failure is removed.
- Capable to “zoom-in” in time
- System can run for long durations without any difficulty.
- Phased array can be assembled on an aircraft or a moving vehicle.
- The system can perform surveillance/tracking of thousands of targets in a given time.
- Remote operations are possible
- The great benefit is graceful degradation. Even some solid state amplifiers are failed in the active phased array, the radar system will work with degraded performance.

1.3 Adaptive Filter

An adaptive filter is a mathematically intensive algorithm that helps in modeling the relationship between two real time signals [11]. These filters can be realized by using arithmetical processing device like a DSP chip/microprocessor or by using field-programmable gate array (FPGA). They can also be realized using VLSI integrated circuits. The specific physical realization of the adaptive filter is independent of its basic operation.

The following characteristics define an adaptive filter:

- What are the signals that are processed.
- How the output signal is computed from input signal.
- How the filter's input-output relationship can be altered by changing different parameters within the structure.
- How the computational algorithm adjusts the parameters from one time instant to the next.

Different communication applications utilize the concept of adaptive weight calculation (AWC). Typical applications include multiple-input multiple-output (MIMO) systems, adaptive beam forming, pre-distortion and equalization [17]. In a most of the applications, over-determined systems of equations is solved. The least squares approach is generally used for approximations. Some of these techniques are Normalized LMS (NLMS), Least Mean Squares (LMS) and Recursive Least Squares (RLS). For fast convergence rates and good numerical properties, the Recursive Least Squares is widely used. But this algorithm is not efficient in hardware implementation and terms of precision as, it requires matrix inversion. More accurate results and efficient architectures can be obtained by the technique of QR decomposition (QRD) based on RLS. [17].

An alternative method to implement the RLS technique is application of QR decomposition and triangularize the input data matrix. When quantization effects are considered, the RLS algorithm based on QR decomposition offers improved numerical behavior that can be used in implementation of systolic arrays [18].

RLS algorithms proposed earlier were based on the QRD technique [6] and mainly focused on information matrix triangularization to avoid the use of inversion of matrix. Due to multiplications for every output sample being of the order of [N^2], the computational requirement was very high. Subsequently, variations of QR-RLS techniques have been proposed with a condensed complexity of computation which is of the order [N]. The fast QR-RLS algorithms are related to the tapped delay line FIR filter realization. Implementation of the algorithms for inverse QRD-RLS algorithm is carried out in Xilinx FPGA.

1.4 Motivation

- The work carried out by the various researchers in this area is having certain limitations. As per the literature survey it is found that most of the work was carried out to address two important issues.
  - Optimized QRD RLS for adaptive weight calculation in terms of area.
  - Optimized processing time.
- The major issue has not been addressed by researchers is that, the complete adaptive beam former VLSI architecture to form multiple receive beams for a phased array radar applications.
- Various researchers have addressed individual module level performance factors and in totality the complete architecture to support for 16 element phased array has not been considered and architecture is not developed which will be suitable for a phased array radar.
- Due to the above reasons it is essential to develop VLSI architecture to support sixteen element phased array radar. The present day requirement of radar is adaptive nulling to be formed in the direction of the jammers i.e. angle of interference.
- It is required to design a scalable architecture which can be extended to a larger number of phased array antennas. This will be a great advantage for futuristic radars to work efficiently in the electronic war scenario. The architecture should be modular and scalable in nature.

1.5 Objectives and Contributions

1.5.1. Thesis Objectives

- **Objective-1**

Design and Realization of a Generic, Modular and scalable VLSI architecture of Adaptive Beam Formation for Sixteen element Phased Array.

- **Objective-2**

The designed architecture should be optimized in terms of speed. The total computation time for one beam should be less than 5 µsec.

- **Objective-3**

The designed architecture should be optimized in terms of Area and should utilize maximum number of DSP slices to handle all the arithmetic operations. The state of the art technology FPGA's should be used to develop the architecture.

- **Objective-4**

The designed VLSI architecture should be validated on the hardware and should meet the timing, functional and interface requirements.

1.5.2. Thesis Contributions – A Summary

- The VLSI architecture of Adaptive Beam Formation for Sixteen element Phased Array is designed which is Generic, Modular and scalable in nature. This architecture is developed in various stages. First the beam formation architecture is designed using the weights which are calculated offline and are stored in a separate memory. Multiple receive beams are formed in the fixed directions around the transmit beams. Q-R Decomposition based RLS algorithm is used for optimal weight calculation. The simulation is carried out in MATLAB to demonstrate multiple receive beams. Second, the modified architecture is designed consisting of digital down conversion, complex multipliers and Inverse Q-R Decomposition RLS module to calculate the weights online. The architecture is simulated in MATLAB and multiple beams have been validated in various look angles in the phased Array Configuration.
- A pipelined and parallel architecture is developed using a systolic array method for optimal weight calculation. The algorithm is mapped in a pipelined sequence of basic computation cells such as Angle processor, Rotation processor, and Weight

processor. These basic cells perform their task in parallel, such that in each clock period all the cells are activated.

  - The Angle processor takes the input data from the input vector matrix and computes the rotation angles. Cosine and sine calculations are carried out and given to the rotation processor to perform the rotation and obtain the new set of values. Angle processor stores the rotated data in its internal memory and uses whenever new data is available.
  - The Rotation processor performs the given rotation on the input data by multiplying it with the cosine and sine from the angle processor to generate the rotated data and stores in its internal memory and passes the output to the cell of next row.
  - Weight processor receives data from rotation processors in its previous row and cosine & sine of the rotation angle from angle processor in its row.

- The initial architecture was designed using the QRD RLS adaptive algorithm. But this algorithm needs back substitution method which will take double the time to arrive at optimal weights. Hence the architecture is modified for Inverse QRD RLS Adaptive algorithm to generate the optimal weights. The architecture is optimized for 4x4 matrix size considering 16 element planar phased array configurations. The optimal time for adaptive weight computation and beam formation achieved is 4.6 u sec.
- The pipelined architecture of IQRD-RLS algorithm implemented for 4x4 matrix size array using multiple FPGA’s. Three FPGA’s are used to form the complete architecture. FPGA1 and FPGA2 architectures remain identical in nature. Each one is interfaced with eight ADC channels for receiving the antenna element data. Total sixteen channel data will arrive simultaneously. It take input from the ADC, and then DDC will make two signals from the ADC input by multiplying the ADC output with the NCO (Numerically Controlled Oscillator) output which is sine and cosine waves, producing two frequency translated signals, one in phase with the input signal and the other is quadrature signal. The DDC is used as a frequency translator. Once the input data matrix is available, based on that Adaptive weight computation using systolic array method, optimal weight will be calculated for each one of the element in the phased array configuration. The optimal weight will be multiplied with the input signal for respective element data in the phased array to form the beam in the desired direction and a null in the interference direction.

- The architecture is optimized to use more number of DSP slices instead of slice registers and slice LUTs. This has resulted in the area optimization of the Adaptive beam formation Architecture. To validate this architecture a proto hardware is used which is having three FPGA’s along with sixteen channel ADC’s on board. A test set up was developed; commands and controls were given from the Remote PC. Real time data captured from the FPGA’s, plotted in MATLAB and validated with the simulated results. The designed architecture has met the functional, timing and interface requirements to form multiple Adaptive Receive beams in the desired direction.

1.6 Thesis Organization

Chapter 1 presents a brief introduction to the Beam formation in phased array Radar. The motivation behind this research work and the thesis objectives and contributions are also mentioned. At the end, organization of the thesis is elaborated.

Chapter 2 describes an elaborated literature survey, carried out by referring various international journals and international conferences. The detailed study has been covered in this chapter.

Chapter 3 explains various algorithms for computation of adaptive weights. The most suitable algorithms are QRD and IQRD RLS for optimal weight calculation. The details of these algorithms are covered.

Chapter 4 includes architecture development for 16 elements Phased Array with MATALB functional Simulation results.

Chapter 5 covers the realizable and optimized VLSI architecture for 16 element array to form one beam within the given time.

Chapter 6 elucidates beam formation using fixed weights and beam formation with adaptive weights architecture development and Implementation on the hardware.

Chapter 7 reveals the results of functional simulation, fixed weight beam formation and adaptive weight beam formation. Comparison of the results using floating point arithmetic and fixed point arithmetic and operations in computing the weights and beam formation architecture.

Chapter 8 presents the conclusion of the research work.

1.7 Summary

In this chapter brief introduction of beam formation for phased array Radar is explained along with the benefits and limitations of phased arrays. The working principle of adaptive filter and various applications of adaptive filters are covered. Particularly for phased array radar application the role of adaptive filter is explained along with various adaptive algorithms. The requirement of optimized adaptive weight computation in terms of processing time and area is covered.

Further the motivation to carry out this novel method of adaptive beam former is explained. The resources optimization and time for the computation of weights are two important parameters are briefed. The thesis objectives are listed and the main contribution is hardware realization of a planar array adaptive beam formation for phased array radar is covered in detail. At the end of the chapter thesis organization is covered mentioning the design challenges, experimental results and conclusion.

In the next chapter a detailed literature survey is covered, mentioning the research work carried out till now and limitations of the same in this area. Many researchers have focused on simulation study of adaptive filters alone but the architecture required for phased array application was not considered. The literature study shows that small section of the entire architecture developed in this thesis was illustrated. Significant information on processing time for complete architecture to form multiple adaptive beams was not illustrated in the literature.

Chapter 2

Literature Review

Chapter 2.

Literature Review

2.1 Phased Array Radar

There are thousands of radiating antenna elements in a modern day active phased radar that makes it a very complex system. Each antenna element consists of dedicated transmitters and receivers. Cluster of sub array transmitters and receiver are spread across the antenna elements. There are dedicated digital communication links that connects all the subsystems. The RF energy will be distributed in the antenna frame with a limited space in the array.

In a Planar Phased array antenna, there is an array of radiating elements on a regular lattice or rectangular grid or triangular grid structure. Several thousand elements will be used to make phased array antenna. In phased-array radars the waveform diversity of MIMO radar is compromised. However, phased-array radar has better performance in terms of side lobe suppression level, computational complexity, and signal-to-noise ratio (SNR) loss with the objective of range and doppler side lobe suppression. This simplifies the design of large scale radar systems.

A good amount of research work has been carried out in this area and observed that various techniques have been identified to reduce the side lobe suppression and to form pencil beam with wider antenna structure. But the adaptive beam formation for larger array dimension is not addressed efficiently by the researchers. The architectures were not developed to meet the present day requirements of the phased array radar.

2.2 Digital Beam formation in phased array Radar

With recent advancements in VLSI technology it has become possible to accommodate several complex DSP algorithms into efficient architectures and realize small high performance Application Specific Integrated Circuits (ASICs) or Field Programmable Gate Array (FPGA) devices. In Digital Beam Forming (DBF) technology antenna technology and digital technology are combined together. There are multiple advantages of using digital methods for beam forming in Radar systems and can achieve high degree of flexibility in performance of the radar by “Beam Pattern Management”. Coherent processing of data collected with an array of sensors enhances coherent wave front, measurement of a propagating wave front relative to ambient background noise and spatially localized interferences in conventional beam former. To achieve this, weighted sensor data are time delayed and summed. The weighted sum of the sensor signals of beam formers is given as output in receive mode due to which, signal dimensions reduce from the number of elements to one. There are various advantages of DBF in receive and transmit ends like high gain, closely placed multiple beam low side lobe levels adaptive nulling, flexible radar power and time management. In a digital beam former system the RF signal is converted into I and Q Channels two streams of binary base band signals representing from each antenna element. Using these I and Q two digital base band signals amplitudes and phases of the signals received at each element of the array can be recovered. Multi-byte A/D converters are available to receive data from antenna, carry out the sampling and provide the output data in digital domain. Faster processing speed and computation power have enabled, the computation of multiple beams digitally using VLSI instead of RF domain. Therefore the deciding factors for number of simultaneous beams are mainly the processor speed and its processing capacity and speed of A/D conversion and its resolution which is ADC number of bits. Input samples are weighted by a complex weighting function adding together to form the desired output. A set of beams that are differently oriented in space are formed as end product of this process with each beam giving access to a number of range and Doppler cells. DBF mainly involves the following key tasks : a) Analog signal is translated into digital domain accurately with the help of high speed ADCs b) Digital mixing, down converting digital filtering is carried out on these high speed samples by DSP methods c) Beams are sent for further processing and plotting using high speed data communication. Researchers have carried out the work on the fixed weight beam formation for communication application and not for radar application. There is a need to develop the VLSI architecture for active phased array application which will greatly help in carrying out the adaptive nulling in the direction of jammers.

2.3 QR decomposition

QR decomposition (QRD) has applications in smart antennas, sonar systems, phased-array radars, adaptive beamforming, channel equalization and, 3G wireless communication, de-noising, echo cancellation and WiMAX. In QR Decomposition a m×n real matrix A as: A=Q x R, where Q is an m×m orthogonal matrix such that QxQ^T=I where I is an identity matrix, and R is an m×n upper triangular matrix. An orthogonal transform of a vector $X=[x_1,...,x_n]^T$ can be defined by the orthogonal matrix Q as: $Y= Q^TX$.

The vector's length: Y=X is not changed in the orthogonal transform. Computational stability and fast convergence are characteristic features of QRD. Many algorithms are available for solving the QRD problem. Some of them are the Givens rotations method, Gram-Schmidt method, Householder transformations, and modified Gram-Schmidt algorithms. Most of the algorithms implementations support a linear i.e one dimension systolic array. Researchers have developed a full two-dimensional (2D) systolic array [114] which is implemented in this work. Unlike all previous work, in Givens a rotation does not avoid divide and square root operations for implementation of QRD. The dynamic range of the input matrix data is accommodated by carrying out all operations floating-point arithmetic. Further, any bit width exponent and mantissa and any size floating-point format, is supported, including standard formats of IEEE. Whereas other work that support only square matrices or tall matrices, QR implementation works for input matrix of any size: short, tall or square matrices. Subject to available hardware resources, the input matrix size can be configured at compile time to virtually any size. As compared to a one dimension systolic array implementation, where latency increases quadratically, latency of two dimension systolic array implementation linearly increases. High throughput and fast clock rate are achieved in QR implementation due to fully pipelined architecture. Also it is easy to scale up the design over multiple FPGAs and future FPGAs.

The main limitations are computational complexity in arriving the optimal weights are not addressed. The speed of operation is an essential factor to be considered for an efficient architecture which has lacked in the previous work done by the researchers. Large size of channel matrix suffers large latencies and low throughput in conventional QR Decomposition hardware.

2.4 Adaptive Phased Array

Target Detection and tracking are specific applications of a radar system. Scanning and tracking or guiding objects is one such crucial application. There are various important requirements of future phased array radar systems which includes high SNR, high sample rate, large size array. To meet this radar system requirement generally seven GHz to thirteen GHz range of RF frequencies will be used for realizing systems. At any given time in the space multiple targets will be available in order to track all the targets simultaneously multiple independent beams are required.

Present day radar systems are using phased array beam forming techniques. Primary concern for design of such systems is functional requirements such as resolution; sensitivity, response time and secondary concern are non-functional requirements such as cost, power consumption [115]. Due to this, it is difficult to find phased array systems that are low-cost and low-power. However, phased array antennas have been very promising in areas like software defined radio and satellite receivers, but their large scale realization cost is very high. Here development of a low-cost, low-power phased array receiver platform is our goal. A scalable architecture can be used for realizing this which is highly flexible and can support a number of applications, in such a way that multiple requirements can be met using same architecture. High performance can be realized using reconfigurable Multiprocessor System-on-Chip (MPSoC) based system architectures. These architectures enable efficient reuse of hardware by reconfiguring parts of an application. As compared to reconfigurable hardware, central processing hardware is used in conventional phased array receivers, that makes the system non scalable less power efficient [116].

2.5 Phased Array Systems and Applications in Radar

Figure 2.1 show radar system modules of a generic phased array system. Signals are received at multiple antennas in a phased array receiver. Antenna Processing (AP) is applied for calibration or equalization purposes after the Radio Frequency (RF) front end for each antenna. By use of equalization process electrical or mechanical distortions are improved for the front-end and the wireless channels. The beam forming unit then combines the signals and creates a resulting signal with maximum sensitivity in desired direction and a minimum sensitivity or a null in other directions. By changing the shape and direction of formed beam that is achieved by changing the delay and gain and the antenna signals before summation is referred as beam steering process. Let’s say a wave front arrives at an angle of incident to the array in a phased array that consists of antennas placed at a distance d apart.

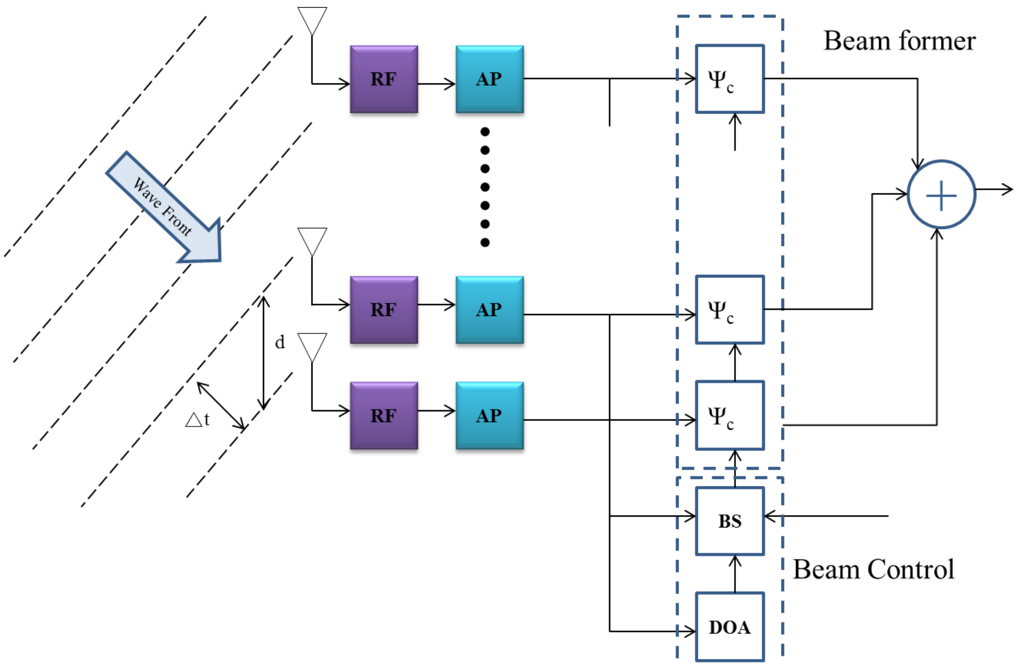


Figure 2-1: Generic Phased Array Receiver

If A time delay t =l x c is observed between the signals where c is the propagation speed of radio waves. The wave front traveling a distance l = d sin (ψ) further to the next antenna, then a phase shift (ψ t) would happen if the signal is a narrowband signal, due to this time delay giving rise to the term ‘phased array’. By correcting the delay [1], the direction of maximum sensitivity steering is achieved.

Fixed array configuration used in traditional adaptive array beam forming can lead to significant performance loss and inefficiencies under different scenarios. A reconfigurable adaptive antenna array strategy becomes particularly important to achieve high signal to noise and interference ratio using fewer antennas as antennas become smaller and cheaper relative to front-ends. Spatial Correlation Coefficient (SCC) can be minimized between the desired signal and the interference and this objective can be achieved.

2.6 Fixed point operations and floating point operations

The VLSI architectures can be designed using either fixed point arithmetic or floating point arithmetic operations. The architecture based on fixed point operations design needs less FPGA resources and computation time is much faster. This also brings the limitations in terms of rounding and truncation errors.

The floating-point arithmetic permits us to symbolize broad range of numbers with constant precision. Finite-point arithmetic symbolizes a decreased range of numbers with appropriate precision absolute. Employing floating precision arithmetic is highly expensive when considered in terms of hardware and results in inefficient architecture, particularly when implemented using FPGA. Alternatively the fixed precision arithmetic representation leads to hardware design that is efficient and also introduces a small amount of error. Here the design employs two's compliment method and the representation of finite precision arithmetic comprise of the sign bit, integer part along with the fractional part. Quantities that find occurrence in the algorithm are characterized with m_n bits designed for the integer along with m bits which are employed for the representation of the fractional bits. Finally, representing a finite precision illustration would entail m_n+m+1 bit, were one bit is utilized as a sign bit representation [80].

QR decomposition needs the usage of arithmetic operations, the calculations required finds an expansion when there is growth in the dimension of the matrix. Large number of calculations is observed during the matrix decomposition process where the orthogonal matrix and upper triangular matrix are obtained. Calculations employed for the decomposition process are the basic arithmetic computation which is a direct approach; complicated computations significantly alter the precision of the entity likely producing the inefficient values during the implementation in the FPGA. Finite-precision arithmetic reduces the obtained precision and thereby introducing two different types of errors which are coined as the round-off and the truncate error. Round-off error [129] makes its presence when the additional bits are required in the outcome than the bits that are reserved usually after the arithmetic computations.

2.7 Papers Referred

1) Researchers have carried out the work on *"A low-complexity high speed QR decomposition implementation for MIMO receivers, published in IEEE Symposium in 2009. In this research a hybrid QRD scheme that uses a combination of multi-dimensional Givens rotations, Householder transformations and the conventional two-dimensional (2D) Givens rotations to both reduce the overall computational complexity and achieve higher execution parallelism. To prove the effectiveness of the proposed QRD scheme, a novel pipelined architecture is presented that uses un-rolled pipelined CORDIC processors iteratively to maximize throughput and resource utilization, while minimizing the gate count. The proposed design achieves the lowest processing time and the highest throughput reported to-date for the same framework. The proposed scheme reduces the overall computational complexity and allows*

higher execution parallelism, and is proved to have the same BER performance as the conventional scheme through simulations. The major limitations of this approach are huge amount FPGA resource utilization".

2) Various researchers have thought of *"Optimal implementation of QR decomposition. A work published in 2011 on Fixed-point CORDIC-based QR decomposition by givens rotations on FPGA, Givens rotations algorithm based QR decomposition systolic array a parallel architecture implemented on FPGA. The proposed architecture adopts a direct mapping by 21 fixed*-point CORDIC-*based process units that can compute the QR decomposition for a 4×4 real matrix. In order to achieve a comprehensive resource and performance evaluation, the computational error analysis, the resource utilized, and speed achieved on Virtex V Field Programmable Gate Arrays, are evaluated with different precision of the intermediate word lengths. The limitations are 32 bit word lengths are not worked out and time and area are not optimized for beam forming application".*

3) Further Researchers have developed *"FPGA implementation of fast QR decomposition based on givens rotation, in 2012 this work was published in fifty fifth IEEE International Midwest Symposium on Circuits and Systems. In this work an enhanced fixed-point hardware design of QR decomposition, particularly optimized for Xilinx FPGAs is evaluated. Givens Rotation method implementation on FPGA is presented. The feedback loop problem is addressed, and a very high throughout is achieved. Further work is expected on large matrix size decomposition as well as using dynamic reconfigurable technology to optimize size and area. The main limitations are it is not a generic architecture for beam formation; uses large number of Resources and computation is also not efficient".*

4) Researchers have worked on *"FPGA based architectures for high performance adaptive FIR filter systems which is published in IEEE International Instrumentation and Measurement Technology Conference in 2013. This work presents a high performance adaptive FIR filter hardware architecture. In particular, the RLS (Recursive Least Square) algorithm for adaptive signal processing is explored based on QR decomposition, which is accomplished by using the Givens Rotation algorithm. The Givens Rotation algorithm is implemented using a systolic array and LUT-based Newton's method. This*

architecture is suitable for high-speed FPGAs or ASIC designs. It also solves the trade-off between throughput and latency issues. As a case study, this QR design is tested using Xilinx XC5VLX110T FPGA. The findings show that the system is capable of running the QR decomposition at up to 200MHz with 56 clock cycles latency. The main limitation of this approach is that, it addresses only the local issue of QR decomposition. This work is not focused on Adaptive beam formation".

5) Researcher have worked on *"Low complexity QR-decomposition architecture using the logarithmic number system, and published in EDA Consortium, 2013.In this work, they have proposed a QR-decomposition hardware implementation that processes complex calculations in the logarithmic number system. The proposed algorithm is simulated with several different configurations in a downlink pre-coding environment for 4x4 and 8x8 multi-antenna wireless communication systems. In addition, the results are compared with the CORDIC-based architectures. In a second step, HDL implementation as well as logical and physical CMOS synthesis are performed. The comparison to actual references highlight this approach as highly efficient in terms of hardware complexity and accuracy. The limitations are, they used wireless communication for data transfer, by using this type of communication the data loss is more and communication speed is very less and it will affect very badly the accuracy of the system".*

6) Researchers have developed on *"FPGA Implementation of Beam forming Algorithm for Terrestrial Radar Application, and published in 2014. Here Beamformer algorithm is developed using LMS algorithm was simulated and implemented hardware using Altera FPGA. Here focus, is to implement adaptive beamforming algorithm known Least Mean Square Algorithm in FPGA. The limitations are, they use LMS algorithm which has very low convergence speed and sample matrix inversion is not possible in this algorithm".*

7) Researchers have simulated on *"FPGA Methodology for Power Analysis of Embedded Adaptive Beamforming IEEE 2015. In this work, they proposed an FPGA-based methodology for the analysis, modeling and prediction of power dissipation in embedded array signal processing systems containing adaptive beamforming components. This FPGA-based methodology enables the*

exploration of the adaptive beamforming design space in terms of power, timing, overhead, arithmetic precision and computational resources. A distinct feature of this methodology is that it enables such design-space exploration in real-time and on actual received waveforms. They describe a specific implementation of this methodology using a hardware prototype based on Xilinx's Virtex 7 FPGA. They use this prototype to explore the design space of a four-channel Least-Mean-Squares (LMS) beamformer. The main result of this exploration is the selection of an adaptive algorithm design point that represents the best tradeoff between parameter convergence, machine precision and energy-efficiency for the embedded array signal processor. The Limitations are, they used LMS algorithm, this algorithm convergence speed is very low compare to QRD RLS and Sample matrix inversion is not possible in this algorithm".

8) Researchers have developed *"a high performance adaptive FIR filter hardware architecture. In particular, the RLS (Recursive Least Square) algorithm for adaptive signal processing is explored based on QR decomposition, which is accomplished by using the Givens Rotation algorithm. The Givens Rotation algorithm is implemented using a systolic array and LUT-based Newton's method. This architecture is suitable for high-speed FPGAs or ASIC designs. It also solves the tradeoff between throughput and latency issues. As a case study, this QR design is tested using Xilinx XC5VLX110T FPGA. The findings show that the system is capable of running the QR decomposition at up to 200MHz with 56 clock cycles latency".*

2.8 Summary

In this chapter a detailed literature survey is covered, mentioning the research work carried out till now and limitations of the same in this area. Many researchers have focused on simulation study of adaptive filters alone but the architecture required for phased array application was not considered. The literature study shows that small section of the entire architecture developed in this thesis was illustrated. Significant information on processing time for complete architecture to form multiple adaptive beams was not illustrated in the literature.

In the next chapter adaptive filter multiple algorithms were covered along with the detailed structure of adaptive filter, convergence rate and computational aspects. The recursive least square (RLS) is mentioned along with conventional QRD-RLS and Inverse QRD-RLS mathematical model.

Chapter 3

Adaptive Filter Algorithms

Chapter 3.

Adaptive Filter Algorithms

3.1 Introduction

The Research area of advanced signal processing and specifically adaptive signal processing has seen a vast development in the last few decades, because of advancement in the technology for realization of the advanced algorithms on the hardware. Large numbers of problems have been solved using these algorithms. These problems include noise cancellation, echo cancellation, signal prediction, channel equalization and adaptive arrays.

There is a close relationship of Adaptive filtering algorithms with regulating mechanism for the filter coefficients with the traditional optimization techniques [69]. However, in the classical approach, all computations are carried out in an off-line method. Due to its real-time dynamic characteristic, an adaptive filter tracks the optimum behavior of a slowly varying environment. There are wide varieties of adaptive algorithms which can be compared based on, following aspects.

- **Computational aspects:** Computational complexity is an important factor for the adaptive filter performance. Dynamically the weight vector needs to be calculated in the real time scenario. The weight vectors should be obtained with minimum number of operations and should be optimal in nature. Also the errors generated during the computation should be as small as possible.

- **Filter structure:** The transfer function gives the relationship between input and output of the adaptive filter. The delay line filter is easy for implementation, simple and efficient. The computational complexity and speed of adaptation with respect changing environment is an important aspect of the adaptive filter.

- **Rate of convergence, tracking and maladjustment:** The weight vectors of a filter will converge in a faster way if the environment is noiseless other way the convergence will be slower if the environment is very noise. The weights will be near to optimum value, but they are how close to the optimum will define the maladjustment. The algorithm convergence speed in non-stationary environment is related to the adaptive filter tracking ability.

In designing adaptive filters for radar applications, recursive least squares (RLS) and constrained recursive least squares (CRLS) [10, 4] algorithms were the promising methods compared to least mean squares (LMS) algorithm due to their fast convergence rate. CRLS and RLS algorithm uses direct inversion input data matrix. It has two major disadvantages. One is that this method has undesirable numerical characteristics when complex covariance matrix is ill conditioned. Another disadvantage is that the RLS and CRLS algorithms cannot be implemented as parallel and pipelined array processors for real time signal processing applications [12].

The adaptive filter has wide range of applications like system identification, channel identification, plant identification, echo cancellation, acoustic echo cancellation, and adaptive noise cancellation. This adaptive filter is used for channel identification where a beam is formed for phased array radar application. All adaptive filters have the same general parts; an input x(n), a desired result d(n), an output y(n), an adaptive transfer function w(n), and an error signal e(n) which is the difference between the desired output u(n) and the actual output y(n) as shown in Figure 3.1. In addition to these parts, the system identification and the inverse system configurations have an unknown linear system u (n) that can receive an input and give a linear output to the given input.

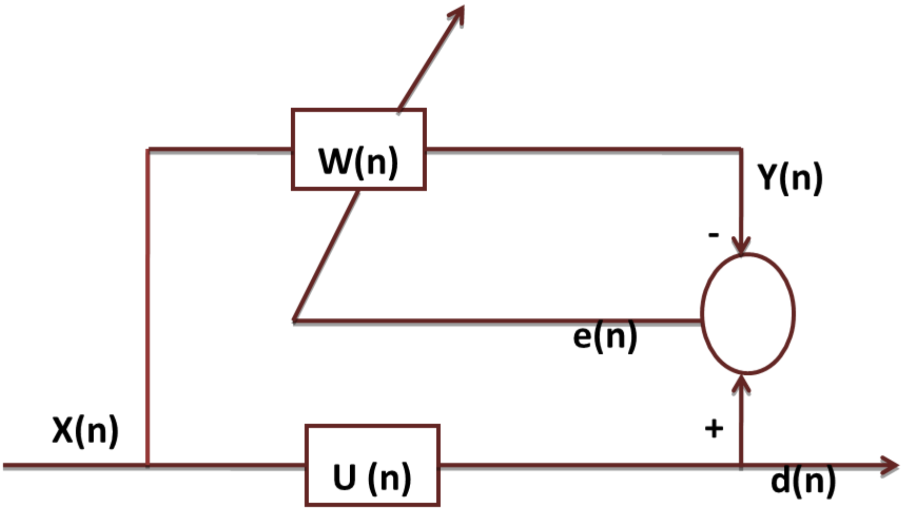


Figure 3-1: Basic adaptive filter structure

3.2 Recursive Least Squares approach

The adaptive filter operates on recursive manner and optimizes the weighted sum of the squared estimation errors; hence it is known as recursive least-squares (RLS) adaptive filters [12]. The RLS finds the precise solution for every iteration [2]. Let us consider the exponentially-weighted RLS adaptive filter [19] and derive the optimal solution.

Let W(*n*) denote the coefficient vector of the adaptive filter at time *n*. Even though we have not yet formulated the problem, assume that W(*n*) is the optimal solution of the problem[62]. Let the estimation error at time *k* due to the coefficient vector W(*n*) be $e_n(k)$, *i.e.*,

$$\mathbf{e_n(k) = d(k) - W^T(n)X(k)} \qquad (3\text{-}1)$$

The exponentially-weighted RLS adaptive filter selects the coefficient vector W(*n*) so as to minimize the exponentially-weighted sum of the squared errors given by

$$\boldsymbol{J(n) = \sum_{k=1}^{n} \lambda^{n-k}\, e_n^2(k)} \qquad (3\text{-}2)$$

Finding the optimal solution is an easy proposition. Substituting “Eq.(3.1)” for e_n into “Eq.(3-2)” and expanding gives

$$\boldsymbol{J(n) = \sum_{k=1}^{n} \lambda^{n-k}\, d^2(k) + W^T(n)\{\lambda^{n-k}X(k)X^T(k)\}W(n) - 2\{\lambda^{n-k}d(k)X^T(k)\}W(n)} \qquad (3\text{-}3)$$

J(*n*) is a quadratic function of the coefficients, it has a unique minimum whenever the matrix within brackets in the second term on the right-hand-side of “Eq.(3.3)” is positive definite. In most cases, the matrix

$$\boldsymbol{\sum_{k=1}^{n} \lambda^{n-k} X(k)}\mathbf{X^T(k)} \qquad (3\text{-}4)$$

is positive definite1 for *n* >*L*. The optimal coefficient vector can be determined from “Eq.(3-4)” by differentiating *J*(*n*) with respect to W(*n*) and setting the resulting vector equal to the zero vector. This operation gives

$$\boldsymbol{2\hat{R}_{xx}(n)W(n) = \hat{P}_{xd}(n)} \qquad (3\text{-}3)$$

where $\hat{R}_{xx}(n)$ and $\hat{P}_{xd}(n)$ are the exponentially weighted least-squares estimates of the autocorrelation matrix of X(*n*) and the cross-correlation vector of X(*n*) and *d*(*n*); respectively. These quantities are defined as

$$\boldsymbol{\hat{R}_{xx}(n) = \lambda^{n-k}X(k)X^T(k)} \qquad (3\text{-}4)$$

and

$$\mathbf{\widehat{P}_{xd}(n) = \sum_{k=1}^{n} \lambda^{n-k} d(k) X^{T}(k)} \qquad (3\text{-}5)$$

Respectively, Assuming that $\widehat{R}_{xx}(n)$ is not a singular matrix, we can solve for W(n) to get

$$\boldsymbol{W(n) = \widehat{R}_{xx}^{-1}(n)\widehat{P}_{xd}(n)} \qquad (3\text{-}6)$$

The equation (3-6), solves the least-squares estimation problem and it is a computationally expensive solution.

3.3 Conventional QRD-RLS Algorithm

Conventional QRD-RLS implementation involves converting the input data matrix to the upper triangular matrix using QR-Decomposition technique in pipelined manner and then performing back substitution to generate the weights. The back substitution procedure is essentially a non-pipelined procedure because the calculation of weights starts by using the last row of upper triangular data matrix whereas the first row of the data matrix gets updated first when new data arrives [12,20].

QR matrix decomposition (QRD), sometimes referred to as orthogonal matrix triangularization, is the decomposition of a matrix (A) into an orthogonal matrix (Q) and an upper triangular matrix (R).

Consider the following equation:

$$\mathbf{AY = Z} \qquad (3\text{-}7)$$

Where, A, Y and Z are matrices, A is of order N × N, Y and Z are a column vectors of order N ×1, A and Z are known; Y is unknown.

The objective is to determine the N different unknowns in the Y matrix. Performing QRD (substituting QR for A) results in:

$$(QR)Y = Z \qquad (3\text{-}8)$$

Moving Q to the right hand side of the equation gives:

$$RY = Q^{-1} Z \qquad (3\text{-}9)$$

Q is an orthogonal matrix, thus Q^{-1} is equal to the complex conjugate transpose of Q. This operation requires minimal resources to perform in hardware. So:

$$RY = Z' \tag{3-10}$$

Where:

$$Z' = Q^{-1} Z \tag{3-11}$$

The disadvantage with back substitution process is that we have to wait for calculation of weights till the last row of data matrix gets updated. Therefore back substitution takes more time to generate weights when compared to Inverse QRD-RLS implementation where weights are calculated in pipelined manner.

3.4 Mathematical Model of Inverse QRD-RLS Adaptive Filter

In conventional QRD-RLS algorithm [9,15], to extract coefficients, it is required to wait till the whole systolic array processing is completed on the input data matrix, which is time consuming. Another algorithm Inverse QR decomposition (IQRD-RLS) algorithm updates the weight vector based on the inverse Cholesky factor, this method allows directly obtaining the weight vector instead of multiple stages unlike in conventional QRD-RLS algorithm. In the following, based on the structure of Q_θ *(k)* and on the relations [23, 31, 33, 34]

$$\mathbf{g}(\mathbf{k}) = -\boldsymbol{\gamma}(\mathbf{k})\mathbf{E}(\mathbf{k})\mathbf{f}(\mathbf{k}) \tag{3-12}$$

$$\mathrm{g(k)} = -\gamma(\mathrm{k})\lambda^{-1/2}\mathrm{U}^{-\mathrm{T}}(\mathrm{k}-1)\mathrm{x(k)} \tag{3-13}$$

$$\mathrm{f(k) = U{-}T(k)\ x(k)} \tag{3-14}$$

$$\mathrm{E(k)} = \lambda^{1/2}\mathrm{U}^{-\mathrm{T}}(\mathrm{k})\mathrm{U}^{\mathrm{T}}(\mathrm{k}-1) \tag{3-15}$$

Here the IQRD-RLS algorithm presented, starting from the RLS solution $w(k)= R^{-1}(k)\ p(k)$ with $R(k)$ and $p(k)$, where $R(k)=X^T(k)X(k)$ is the $(N+1)\times(N+1)$ input-data deterministic autocorrelation matrix, and $p(k)= X^T(k)d(k)$ is the $(N+1)\times 1$ deterministic cross-correlation vector. For that,

$$X(k)=\begin{bmatrix} x^T(k) \\ \lambda^{\frac{1}{2}}X(k-1) \end{bmatrix}$$

Instead of *X(k),* after some manipulations, we can show that

w(k)= w(k−1)+e(k)U−1(k)U−T(k)x(k) (3-16)

Where $e(k)=d(k)-x^T(k)w(k-1)$ is the apriori error and the term multiplying this variable is known as the Kalman Gain. Also note, knowing that $R(k)= U^T(k)U(k)$.Since we know that $Q_\theta\ (k)$ is unitary, if we post-multiply this matrix by its first row transposed, it follows that [35, 37];

$$Q_\theta\ (k)\begin{bmatrix}\boldsymbol{\gamma}(\mathbf{k})\\ \mathbf{g}(\mathbf{k})\end{bmatrix}=\begin{bmatrix}\mathbf{1}\\ \mathbf{0}\end{bmatrix} \qquad 3\text{-}17)$$

We have $g(k)=-\gamma(k)\lambda^{-1/2}U^{-T}(k-1)x(k)$. For convenience, we define $a(k)=-\gamma^{-1}(k)g(k)=\lambda^{-1/2}U^{-T}(k-1)x(k)$

$$Q_\theta(k)\begin{bmatrix}\mathbf{1}\\ -\boldsymbol{a}(\boldsymbol{k})\end{bmatrix}= \quad \begin{bmatrix}\boldsymbol{\gamma}-\mathbf{1}(\boldsymbol{k})\\ \mathbf{0}\end{bmatrix} \qquad (3\text{-}18)$$

This expression, if we know a(k), provides $Q_\theta(k)$. At this point, it is relevant to observe that $\lambda^{-1/2}E(k)U^{-T}(k-1)= U^{-T}(k)$, suggests that $U^{-T}(k-1)$ can be updated with the same matrix that updates $U(k-1)$. In fact, if we rotate $[0 \quad \lambda^{-1/2}U^{-T}(k-1)]^T$ with

$Q_\theta\ (k)$, we obtain

$$\begin{bmatrix}\boldsymbol{\gamma}(\boldsymbol{k}) & \boldsymbol{g}^{\boldsymbol{t}}(\boldsymbol{k})\\ \boldsymbol{f}(\boldsymbol{k}) & \boldsymbol{E}(\boldsymbol{k})\end{bmatrix}\begin{bmatrix}\mathbf{0}^{\boldsymbol{T}}\\ \boldsymbol{\lambda}^{-\mathbf{1}/\mathbf{2}}\boldsymbol{U}^{-\boldsymbol{T}}(\boldsymbol{k}-\mathbf{1})\end{bmatrix}=\begin{bmatrix}\boldsymbol{\lambda}^{-\mathbf{1}/\mathbf{2}}\mathbf{g}^{\mathbf{T}}\mathbf{U}^{-\mathbf{T}}(\mathbf{k}-\mathbf{1})\\ \mathbf{U}^{-\mathbf{T}}(\mathbf{k})\end{bmatrix} \qquad (3\text{-}19)$$

For convenience, we define $u\ (k) = \lambda^{-1/2}U^{-1}(k-1)g(k)$. Using vector *a(k),* this vector can be expressed as

$u(k)=-\lambda^{-1/2}\gamma(k)U^{-1}(k-1)a(k)$ or, as $u(k)= -\gamma^{-1}(k)U^{-1}(k)U^{-T}(k)x(k)$.

Finally, with this last equation, w(k) can be rewritten as $w(k)= w(k-1)-e(k)\gamma(k)u(k)$ where the Kalman vector is now expressed as $-\gamma(k)u(k)$.

By combining Equation 3.18 and Equation 3.19 in one single equation, we have

$$\begin{bmatrix} \boldsymbol{1}/_{\boldsymbol{\gamma}(\boldsymbol{k})} & \boldsymbol{u}^{\boldsymbol{T}}(\boldsymbol{k}) \\ \boldsymbol{0} & \boldsymbol{U}^{-\boldsymbol{T}}(\boldsymbol{k}) \end{bmatrix} = \boldsymbol{Q}_{\boldsymbol{\theta}}(\boldsymbol{k}) \begin{bmatrix} \boldsymbol{1} & \boldsymbol{0}^{\boldsymbol{T}} \\ -\boldsymbol{a}(\boldsymbol{k}) & \boldsymbol{\lambda}^{-\frac{1}{2}}\boldsymbol{U}^{-\boldsymbol{T}}(\boldsymbol{k}-\boldsymbol{1}) \end{bmatrix} \qquad (3\text{-}20)$$

The above equation is an important relation of the inverse QRD-RLS algorithm.

3.5 Summary

In this chapter adaptive filter multiple algorithms were covered along with the detailed structure of adaptive filter, convergence rate and computational aspects. The recursive least square (RLS) is mentioned along with Conventional QRD-RLS and Inverse QRD-RLS mathematical model.

In the next chapter brief introduction of adaptive digital beam former system is mentioned along with the mathematical model of phased array antenna. A detailed mathematical evaluation is carried out for planar phased array and linear phased array. The QRD RLS application for multiple beam formation in phased array is discussed in detail. The adaptive beam former architecture implementation using systolic array is brought out in detail and automatic update of the weights using this method is discussed.

Chapter 4

Design and Modeling of Adaptive Digital Beam Former

Chapter 4.

Design and Modeling of Adaptive Digital Beam Former System

4.1 Introduction

A mathematical model of adaptive beam former architecture is developed for a sixteen element phased array configuration. The architecture consists of various modules and there are three different QR decomposition methods: Gram-Schmidt orthogonalization, Givens Rotations and Householder reflections. Givens rotation is preferred because of its stability and accuracy. Givens rotation lends itself easily to a systolic array architecture using CORDIC blocks which makes an efficient hardware implementation. Therefore, it is widely used for hardware implementation [12, 41, 43].

A wide variety of computationally intensive applications are moving from Digital Signal Processors (DSPs) to Field Programmable Gate Arrays (FPGAs) because of more efficient implementations. Moreover, FPGAs are a flexible, cost effective alternative to Application Specific Integrated Circuits (ASICs). FPGAs are perfect platforms for arithmetic operations such as matrix decomposition as they provide powerful computational architectural features [63, 70, 71].

4.2 Phased Array Antenna Model

A plane wave front is generated from a group of elementary spherical waves is the principle of phased array. The spherical waves are approximately realized by elementary antenna elements with near omni-directional characteristic.

A phased array radar consists of n elements spaced equally with distance of ‘d’. If a plane wave is incident on the array with an angle φ then the n^th element in the phased array can be denoted as,

$$I_n = A_n e^{j\frac{2\pi}{\lambda} n d \sin(\varphi)} \tag{4-1}$$

λ => Wavelength   n=> Number of Radiating elements in the array

At any given time the phase difference θ with respect to the reference input A0 (n). The steering vector for the linear array is given by:

$$S(\theta) = [\ 1\ e^{-j\theta} e^{-2j\theta} e^{-3j\theta}\ ---\ e^{-Nj\theta}] \quad (4\text{-}2)$$

where, $\theta = \frac{2\pi}{\lambda} d \sin(\varphi)$ and φ is the angle of incidence.

4.3 Planar Phased Array Theoretical Background For Beam Formation

Consider that the elements of the planar array antenna are placed on a rectangular grid structure in the matrix form KxL where K is the number of rows and L is the number of columns with dx the standard distance between the rows and dy the standard distance between the columns.

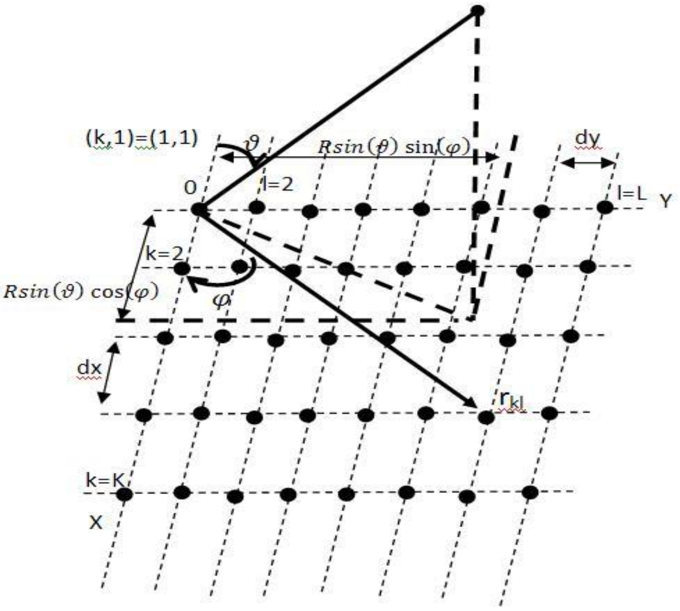


Figure 4-1: Array Matrix of K × L elements represented in Cartesian coordinate system.

The path-length differences between the elements can be obtained by projecting the plane wave direction onto the planar element position vector. For a linear array the phase differences are shown below in Figure 4-2.

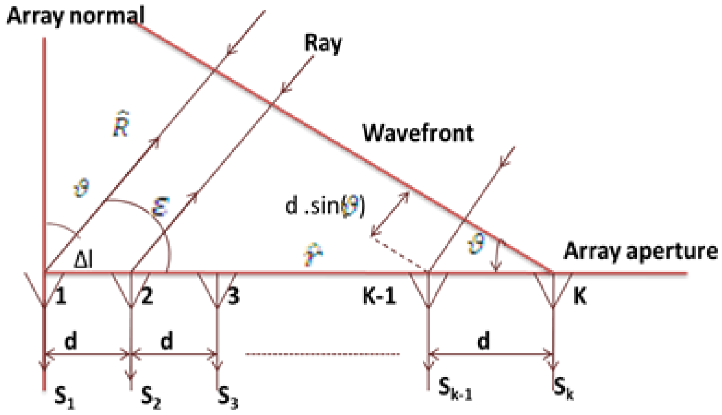


Figure 4-2: A linear array of K elements with d as the inter-element distance.

It is convenient to consider the reference as angle as ψ =0 deg at an element K for the purpose of analysis. The phased array principle works on phase difference between the elements and not on the absolute values. Hence any element can be a reference element. The phase difference can be computed accordingly. Planar array representation is much easier for phase computation as shown in Figure 4-1. Considering the (KxL) matrix then it can be rewritten for a linear array with 'l' as the phase reference.

In a linear array path-length difference between two adjacent elements may be calculated as the dot product of the plane wave direction (R) and the element position vector (r),

$$\Delta l = \hat{R} \, . \, r \ = \hat{R}.\hat{r}d = d\cos(\varepsilon) = \ d\sin\,(\vartheta) \qquad (4\text{-}3)$$

The phase difference, Δψ, between two neighboring elements will be,

$$\Delta\psi = k0.d\,\sin\,(\vartheta) \qquad (4\text{-}4)$$

and the phase of element k relative to element 1, ψk, is given by

$$\psi k = k0(k-1)d\,\sin(\vartheta), \text{for } k = (1, 2,\ldots,K) \qquad (4\text{-}5)$$

where k0=2π/λ0.

Equation (4-5) is easily translated to the planar array antenna configuration from inspection of Figure 4-1. Assume that every element is not weighted (aij = 1) and not phased (ψij =0). For element (k, l), the difference in path length for the plane wave to this element, relative to the (1, 1) -element, Δl kl, is

$$\Delta l_{kl} = R.r_{kl} = (k-1)d_x\hat{R}.\hat{u}_x \ + (l-1)d_y\hat{R}.\hat{u}_y$$

$$= (k-1)d_x\sin(\vartheta)\cos(\varphi) + (l-1)d_y\sin(\vartheta)\sin(\varphi). \qquad (4\text{-}6)$$

The phase of element (k, l) relative to the (1, 1)-element, ψkl, is then given by

$$\Psi_{\mathbf{kl}} = \ k_0(k-1)dx\sin(\vartheta)\cos(\varphi) + (l-1)d_y\sin(\vartheta)\sin(\varphi)$$

Resulting in the planar array radiation pattern

$$S(\vartheta,\varphi) \mathbf{= S_e}(\boldsymbol{\vartheta},\boldsymbol{\varphi})\sum_{\mathbf{k=1}}^{\mathbf{k}}\sum_{\mathbf{l=1}}^{\mathbf{L}}\mathbf{e}^{\mathbf{j(k_0(k-1)d_x sin(\theta)\cos(\varphi)}} + \mathbf{k_0}((\mathbf{l}-\mathbf{1})\mathbf{d_y}\mathbf{sin}(\boldsymbol{\vartheta})\ \mathbf{sin}(\boldsymbol{\varphi})) \qquad (4\text{-}7)$$

where S(ϑ,φ) is the element radiation pattern.

It is considered that all elements are identical in nature and with very less mutual coupling. Hence the radiation pattern of the planar array is;

$$S(\vartheta,\phi) = \ Se(\vartheta,\varphi)Sa1\ (\vartheta,\varphi)Sa2\ (\vartheta,\varphi) \qquad (4\text{-}8)$$

Where

$$Sa1\ (\vartheta,\varphi)= \ \sum_{l=1}^{L} e^{jk_0(l-1)d_x}\ \sin(\vartheta)\cos(\varphi)$$

$$(4\text{-}9)$$

is the array factor for the linear array in the x-direction and

$$Sa2\ (\vartheta,\varphi)= \ \sum_{l=1}^{L} e^{jk_0(l-1)d_y}\ \sin(\vartheta)\sin(\varphi)$$

$$(4\text{-}10)$$

is the array factor of the linear array in the y-direction.

The planar array factor, Sa (ϑ, φ), can be represented as the product of two linear array factors, Sa(ϑ, φ)= Sa1(ϑ, φ) and Sa2(ϑ, φ).

The elements are arranged in rectangular grid and the directivity increased in both principal planes of the array antenna.

4.4 QRD-RLS Algorithm

Any real square matrix A may be decomposed as

$$\mathrm{A} = \mathrm{Q} * \mathrm{R} \tag{4-11}$$

where, Q is an orthogonal matrix (its column are orthogonal unit vectors meaning $Q^T Q = I$) and R is an upper triangular matrix.

Weights w (n) corresponding to each antenna element at time t_n can be found out using RLS algorithm as follows.

$$\mathrm{R_{xx}(n)w(n)} + \mathrm{p(n)} = 0 \tag{4-12}$$

Here, Rxx is the (M-1)*(M-1) data covariance matrix X, where M is the number of sensors, and p(n) is the (M-1) element cross correlation vector.

The QR decomposition technique can be used in solving matrices, for evaluating the weights.

The QR decomposition can be applied to the least squares problem given above as:

$$\mathrm{Q(n)X(n)} = \binom{\mathrm{R(n)}}{0} \tag{4-13}$$

where, R (n) and Q (n) represents (M -1)*(M -1) upper triangular matrix and (M -1)*(M -1) orthogonal matrix respectively. Since Q(n) is an orthogonal matrix the residue vector e(n) can be estimated as:

$$\|\mathrm{e(n)}\| = \left\| \binom{\mathrm{R(n)}}{0} \mathrm{w(n)} + \binom{\mathrm{u(n)}}{\mathrm{v(n)}} \right\| \tag{4-14}$$

Where,

$$\binom{\mathrm{u(n)}}{\mathrm{v(n)}} = \mathrm{Q(n)d(n)} \tag{4-15}$$

The d(n) indicates the reference data of the systolic array. The RLS weight vector that minimizes error ||e(n)|| can be calculated as per (4-16).

The conventional QRD-RLS algorithm involves calculation of the weights by the process of back substitution. After reducing the input data matrix to the upper triangular matrix, the back substitution process starts.

But the disadvantage of the back substitution process is that it is a non-pipelined process. Thus time required for the calculation of the weight vector by QRD-RLS algorithm mainly depends on the back substitution.

This disadvantage is overcome by the Inverse QRD-RLS algorithm [2]. In this algorithm the inverse of the input data matrix is updated along with the original input data matrix. Thus it obviates the need for the back substitution process and gives the weight vector at each instant in pipelined manner.

4.5 Implementation using Systolic Array Method

The systolic array implementation method for a given algorithm contains mapping the algorithm in a parallel and pipelined method of basic algorithm cells as shown in Figure 4-3. The cells of the systolic array work in parallel in such a way that for every clock period all the cells will be operational [5, 9, and 13].

The principle of Givens rotation works on two important stages. The first stage is the computation of the cosine and sine, every elements of the rotation matrix. Next stage is the use of the rotation matrix to input data. So the basic elements angle processor and rotation processor are used for systolic array implementation. The sine and cosine are computed by angle processor and transfers to respective outputs. The partial product of cosines meant for the error signal is carried out in rotation processor. It also computes the rotation between the incoming data from input1 with the internal processed data of the input data matrix U (k) and this result will be communicated to next stage rotation processor. The rotation processor also updates the elements of U(k) and handovers the cosine and sine values to the adjacent cell on the left side of the matrix[15,21,28].

At every moment an element of the input matrix U(k) is stored in the basic cells. These deposited data are arriving at different intervals of time and in the beginning they all will be forced to zero. The left side processing units store the elements of data vector d (k), which are initialized to zero and updated on regular intervals with respect to every clock.

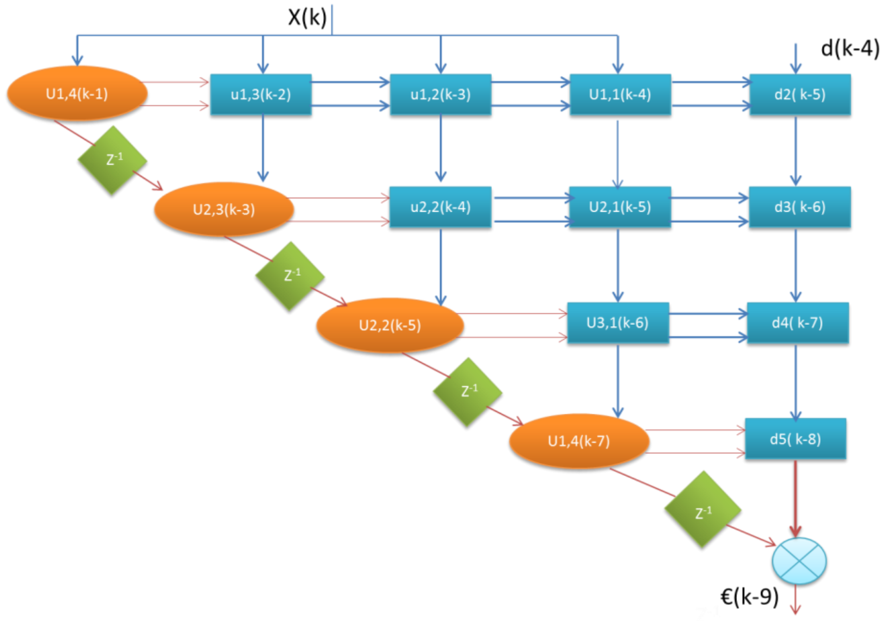


Figure 4-3: Parallel and pipelined data flow in systolic array.

The column on the left hand side carries out the rotation and deposits the rotated data of the desired signal vector and this is required to compute the error signal during further processing. The architecture is pipelined approach, where the outputs of each cell are calculated with the actual clock and the data are made available to the next clock. Also the adjacent cells on the left hand are performing the computations concerned to previous iteration and the adjacent cells on the right hand are performing the advance computations for the next iteration [8, 28, 29].

The basic cells in every row in the array does a Given rotation between $\lambda U(k-1)$ and a vector with the incoming data X(k) from the array elements. The first row of the array carries out zeroing of the last member of the latest incoming data vector of X(k). This result obtained will be moved to the second row in the antenna array. The similar method is followed for zeroing the second to last member in the input signal which is already rotated. The process will continue in the next rows by eliminating the remaining elements in between vectors $X_i(k)$, by Givens rotations principle. The angle of rotation will be computed by the processors like angle and weight cells which will be passed transferred to every row to perform rotations.

The other two methods of QR decomposition are Gram-Schmidt orthogonalization and Householder reflections. The Householder scheme of QR decomposition is essentially very simple and numerically steady due to the use of reflections as the mechanism for producing zeroes in the R matrix. But, this scheme needs higher bandwidth and parallel processing is not possible in applications like Adaptive beam formation for Radars, as every reflection that produces a new zero element changes the entirety of both Q and R matrices. The main limitation of Gram–Schmidt is small inaccuracies in the computation of inner products accumulate rapidly and results into loss of orthogonality due to rounding error in the process.

Givens rotation is preferred because of its stability and accuracy. Givens rotation lends itself easily to a systolic array architecture using CORDIC blocks which makes an efficient hardware implementation. Therefore, it is widely used for hardware implementation. In this particular application other two methods needs more time for decomposition as the limit is within five micro second the optimal weight calculation and beam formation should be completed to ensure that the Radar is not jammed by the enemy Jammers in the electronic warfare scenario and to prevent the antenna element data overflow.

4.6 Summary

In this chapter brief introduction of adaptive digital beam former system is mentioned along with the mathematical model of phased array antenna. A detailed mathematical evaluation is carried out for planar phased array and linear phased array. The QRD RLS application for multiple beam formation in phased array is discussed in detail. The adaptive beam former architecture implementation using systolic array is brought out in detail and automatic update of the weights using this method is discussed.

In the next chapter development of adaptive digital beam former architecture is discussed in detail. The FPGA implementation of systolic array consisting of sub modules like angle processor, rotation processor and weight processor are explained along with the detailed internal design. In conventional QRD RLS, the optimal weight vectors arrive in two steps viz; QR decomposition and back substitution. This brings the major limitation in achieving the processing time for the phased array application. The Inverse QRD RLS method brings the advantage over the conventional method and process the matrix inversion simultaneously and generating the optimal weight vectors. The fixed weight beam former architecture is discussed along with benefits and limitations. The adaptive weight calculation for sixteen element planar phased array architecture is explained and the data flow is covered starting from the antenna element level digitization to multiple beam formation. This architecture is developed for floating point operations as well as fixed point operations. The comparison has been brought out listing the benefits and limitations.

Chapter 5

Development of Adaptive Digital Beam Former Architecture

Chapter 5.

Development of Adaptive Digital Beam Former Architecture

5.1 Introduction

The motivation for development of the adaptive beam former architecture for Radar application is to provide multiple spatial channels in the digital domain for advanced signal processing algorithm. This architecture provides adaptive nulling of jammers, multiple simultaneous receive beams, angle estimation with high resolution and larger dynamic range of the phased array radar system.

The architecture developed uses a systolic array implementation of an adaptive algorithm. This method employs mapping the algorithm in a parallel and pipelined sequence manner of calculation of basic cells. These basic cells perform their task individually in parallel in such a way that for every clock period all the cells are activated [5, 9, and 13] and computation will be carried out simultaneously.

5.2 FPGA implementation of systolic array

The inverse QRD RLS algorithm, hardware implementation is possible in pipelined and parallel architecture, whereas other RLS variants are difficult to implement in pipelined manner. This algorithm is having numerical robustness and provides the coefficient vector at every iteration, without depending on the computationally tedious backward or forward substitution procedures. Conventional QRD-RLS method needs two stages which is inefficient for parallel pipelined array. Inverse QRD-RLS algorithm allows the calculation of the weight vector in completely parallel and pipelined manner [28, 36]. The basic cells of the systolic array are Angle Processor, Rotation Processor and weight processor.

5.2.1 Angle processor:

The Angle processor takes the input data from the input vector matrix and computes the rotation angles. The sine and cosine values will be calculated and given to the rotation processor to perform the rotation and get the new set of numerical values. The angle processor is designed to have internal memory to stock the rotated data which further will be used for computation whenever new data arrives as shown in Figure.5-1. All the computations are performed using single precision floating point [44, 45] arithmetic.

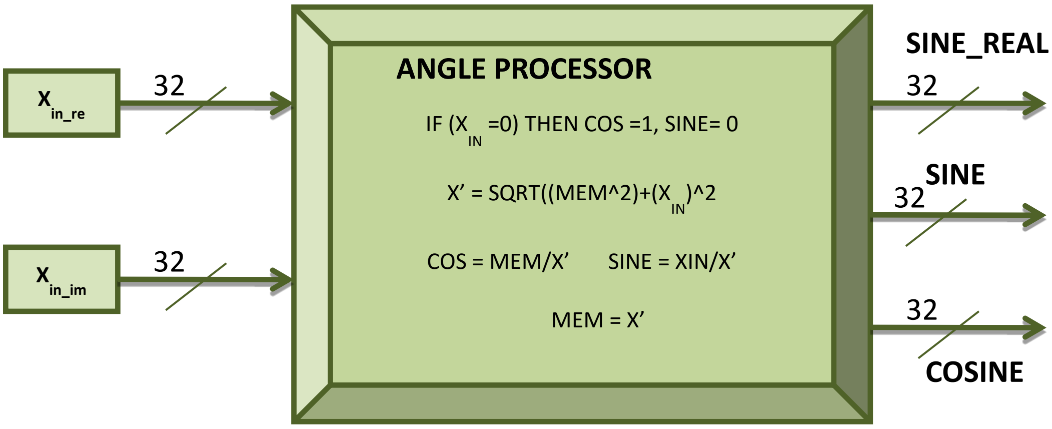


Figure 5-1: Angle Processor cell of Inverse QRD-RLS Algorithm.

5.2.2 Rotation processor:

This cell performs the Givens rotation principle on the input data by multiplying it with the sine and the cosine from the angle processor. It produces the rotated data which is stocked in the internal memory and the output is processed to the cell in the next row as shown in Figure 5-2.

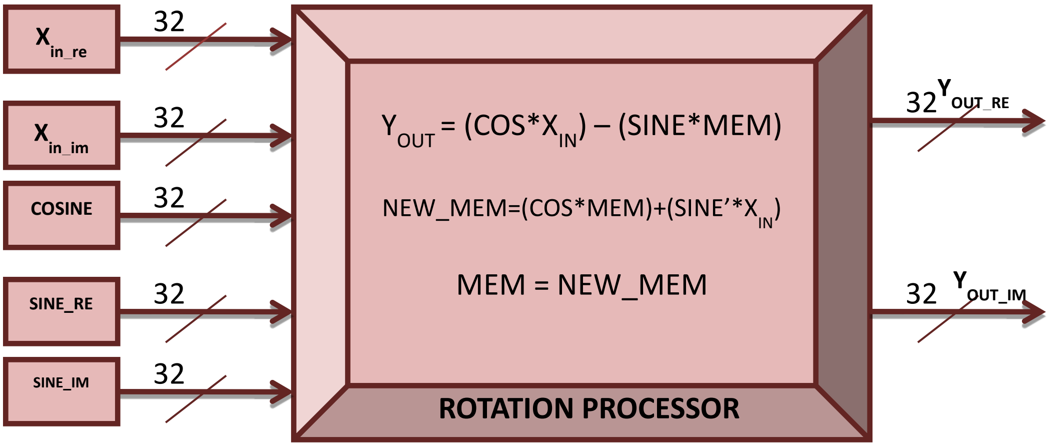


Figure 5-2: Rotation processor of Inverse QRD-RLS Algorithm.

The conventional QRD RLS method of new sets of weight generation is similar to the Inverse QRD RLS method where in the back substitution procedure is eliminated in case of Inverse QRD RLS method. In a parallel pipelined manner another set of cells are required to generate the final optimal weights which are called as weight processors.

5.2.3 Weight processor*:*

This cell receives the data from the rotation processors in its preceding row and the sine and cosine of the rotation angle from the angle processor in its row.

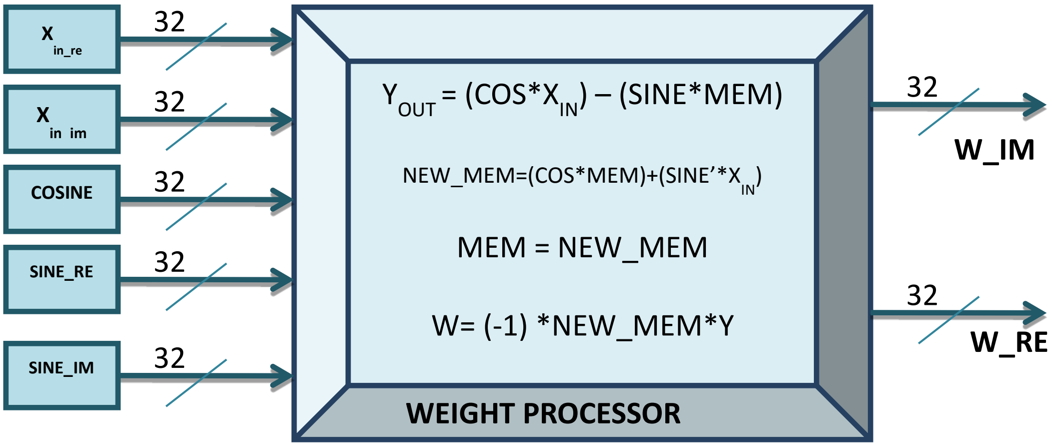


Figure 5-3: Weight processor of Inverse QRD-RLS Algorithm

Some part of the processing is carried out in the weight processor which is similar to the rotation processor operation wherein it rotates the input data by multiplying it with the sine and cosine of the rotation angle as shown in Figure 5-3. All the computation is carried out in single precision floating point arithmetic operations.

$$\mathbf{W}(\mathbf{k}) = -\mathbf{1}(\boldsymbol{COS} * \mathbf{MEM}) + (\mathbf{SINE}' * \mathbf{Xin}) \qquad (5\text{-}1)$$

Along with this operation, the weight processor multiplies the rotated input data value with the scaling factor *(Y)* and then negates the result to generate the final optimal weight vectors. The optimal weights will be calculated with the latency of 4 input samples.

5.3 Adaptive optimal weight computation Methods

There are several adaptive algorithms such as LMS and its variants, RLS and its variants. The most suitable for adaptive beam formation for phased array radar application are QRD RLS and Inverse QRD RLS algorithms.

5.4 Adaptive weight computation using QRD RLS method

The Conventional QRD-RLS implementation involves consists of basic cells viz; angle processor, rotation processor. Rotation processor and Angle processor will receive the data from the input data matrix which will be converted to upper triangular matrix using QR-Decomposition technique in pipelined manner. The lower rows of the matrix will be made zeros and all the required data will be on the upper triangular matrix. After the conversion to upper triangular matrix back substitution is performed to generate the optimal weights as shown in Figure 5-4. This is basically a two stage operation is carried out viz; conversion of input data matrix to upper triangular matrix and back substitution. The main drawback with back substitution process is that needs to wait for calculation of weights till the last row of data matrix gets updated. Therefore back substitution takes more time to generate weights than compared to Inverse QRD-RLS implementation where weights are calculated in pipelined manner.

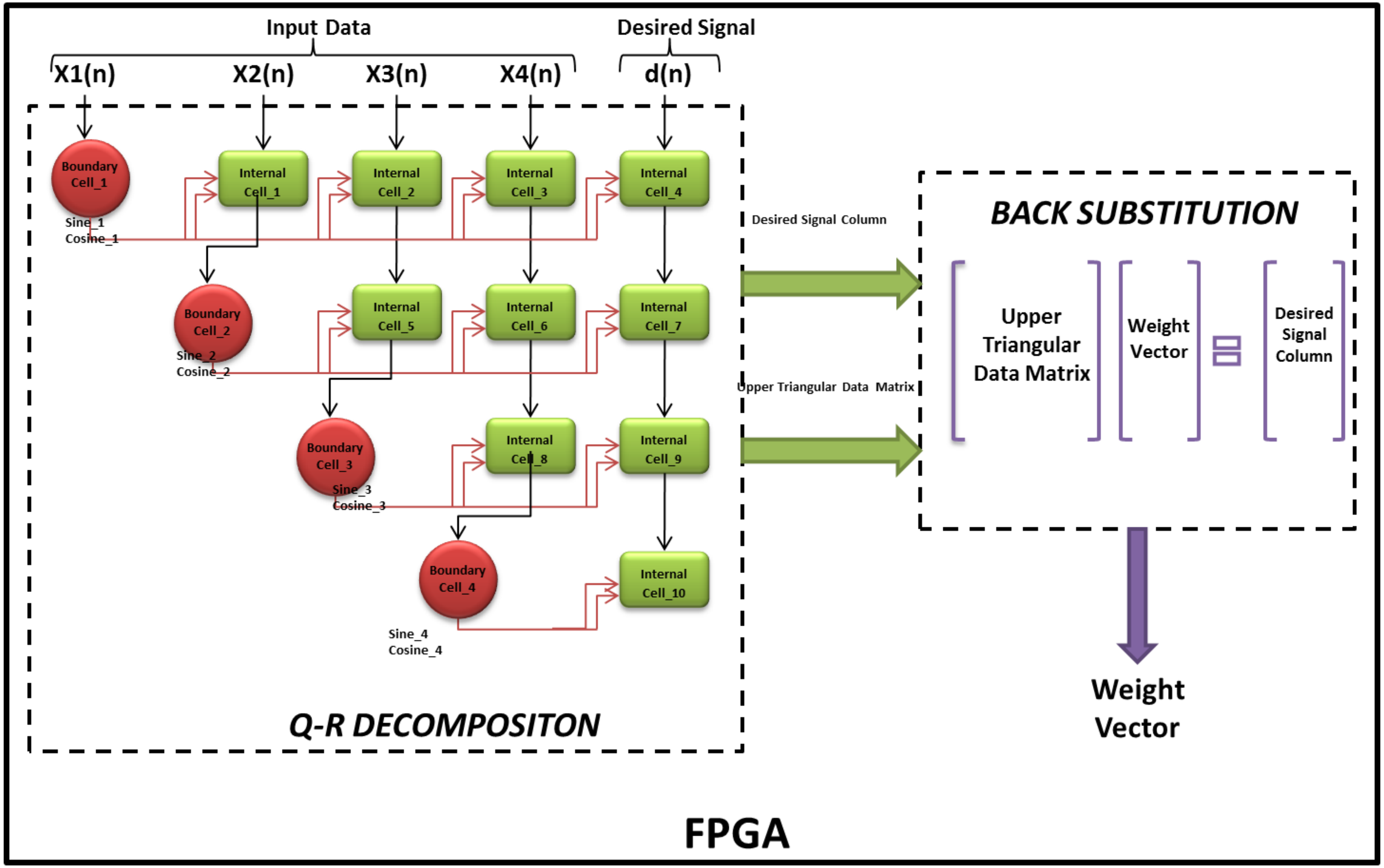


Figure 5-4: Data flow of Inverse QRD-RLS algorithm for optimal weight calculation.

5.5 Adaptive weight computation using IQRD RLS method

The inverse QR decomposition recursive least squares method is the most optimized for computation of adaptive weights for beam formation for phased array Radar. The systolic array mapping of Inverse QRD RLS is shown in Figure 5-5. The realization consists of mapping the algorithm in a pipelined sequence to basic computation cells. These basic cells perform their task in individually parallel, such that for every clock period each one of the cells is activated to perform the pre-designated task. Two steps are required for Givens rotation method: first is computation of the elements of rotation matrix consisting of sine and cosine and then the application of the rotation matrix to given data after that weight calculation. Therefore, the computational elements required for the array signal processor implementation of the IQRD-RLS algorithm [15] introduced are the Angle cell (Angle Processor), Balancing cell (Rotation Processor) and Weighing cell (Weight Processor) [38, 40].

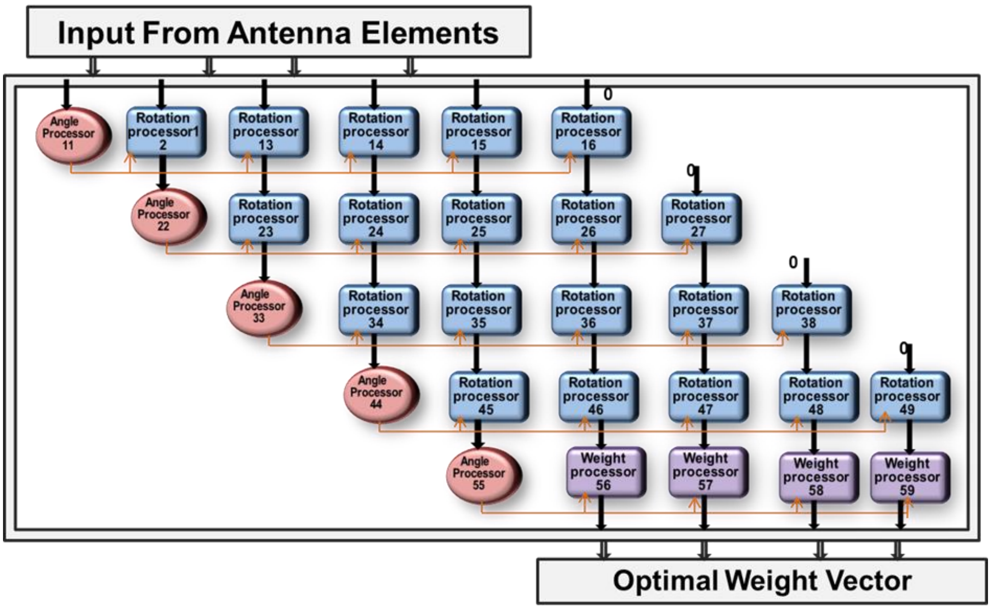


Figure 5-5: Pipelined Architecture of IQRD-RLS dataflow for VLSI Implementation.

Inverse QRD-RLS algorithm allows the calculation of the weight vector in completely pipelined manner.

A. **Angle cell:** Angle cell calculates the rotation angles based on the input signal arrives and passes these calculated values to the balancing cells which are on the right side [45].

B. Balancing cell: Balancing cells does the balancing of the rest of data input, by multiplying the input from the angle cell with the array input signal. In terms of Givens rotation, it performs the rotation of the data [45].

C. Weighing cell: Weighing cells are intended for calculating the weights from the outputs of the angle cells and the balancing cells. For calculating the weights, it takes the outputs from the balancing cells and performs multiplication with the angle cell outputs of that row, then it weights the result with a weighing factor (K) and negates the results for getting the final weight vector [45]. The signal [1,2] as in

$$\boldsymbol{\epsilon(k) = \epsilon_{q1}(k) \prod_{i=0}^{N} \cos\theta_i(k) = \epsilon_{q1}(k)\gamma(k)} \qquad (5\text{-}2)$$

The balancing cell performs the balancing of the rest data by rotation between the data coming from angle cell with the internal element of the matrix U(l) and transfers the result to next cell. This processor also updates the elements of U(l) and transfers the cosine and sine values to the neighboring cell on the left. Assume that upper triangular matrix *U(k)* arranged below, the rows consisting of the new information data vector as in

$$\mathbf{Q(k)X(k)} = \ \widetilde{\mathbf{Q}}(\mathrm{k})\begin{bmatrix} \mathbf{x(k)} & \mathbf{x(k-1)} & . & \mathbf{x(k-N)} \\ \mathbf{0} & \mathbf{0} & . & \mathbf{0} \\ . & . & . & . \\ \mathbf{0} & \boldsymbol{\lambda}^{\frac{1}{2}}\mathbf{U(k-1)} & . & \mathbf{0} \end{bmatrix} \qquad (5\text{-}3)$$

and

$$\boldsymbol{Q_{\theta}(k)} \begin{bmatrix} \boldsymbol{x^T(k)} \\ \boldsymbol{\lambda^{\frac{1}{2}}U(k-1)} \end{bmatrix} = \boldsymbol{Q'_{\theta N}(K)Q'_{\theta N-1}(K)} \ldots\ldots \boldsymbol{Q'_{\theta I}(K)} \begin{bmatrix} \boldsymbol{x'_i(k)} \\ \boldsymbol{u'_i(k)} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{U(K)} \end{bmatrix} \qquad (5\text{-}4)$$

It can be arranged in such a way that the basic cells in order to compute the rotations of the IQRD-RLS algorithm, with the input signal x(k) arriving in a particular sequence manner.

$$\mathbf{d}(\mathrm{k}) = \begin{bmatrix} \boldsymbol{\epsilon_{q1}(k)} \\ \boldsymbol{d_{q2}(k)} \end{bmatrix} = \boldsymbol{Q_\theta(k)} \begin{bmatrix} \boldsymbol{d(k)} \\ \boldsymbol{\lambda^{\frac{1}{2}} d_{q2}(k-1)} \end{bmatrix} \qquad (5\text{-}5)$$

Every element in the matrix is updated for every clock which will be initially set to zero. The first left half of the array carries out the rotation and the values are stored in memory which will be required for computation of the error signal. The last row in the pipelined architecture contains the weighing cells are intended for calculating the final weights from the Angle Cell and Balancing Cell outputs.

The outputs of each cell are calculated in the present clock period and will be available to the neighboring cells in the next clock period for pipelining architecture. The basic cells in every row in the array does a Given rotation between λU(k−1) and a vector with the incoming data x(k) from the array elements. The first row of the array carries out zeroing of the last member of the latest incoming data vector of x(k). This result obtained will be moved to the second row in the antenna array. The similar method is followed for zeroing the second to last member in the input signal which is already rotated. The process will continue in the next rows by eliminating the remaining elements in between vectors x (k), by Givens rotations principle. Finally the all the zeros are in the lower triangular matrix and required values are in the upper triangular matrix are obtained.

5.6 Fixed weight four element Beam Former architecture to form four simultaneous beams.

The architecture for the beam former is developed in various stages considering the optimization to form multiple receive beams. The Figure 5- 6 shows the data flow of the beam former architecture. It is considered that four element phased array configuration represented in 1, 2, 3 and 4 in the data flow chart.

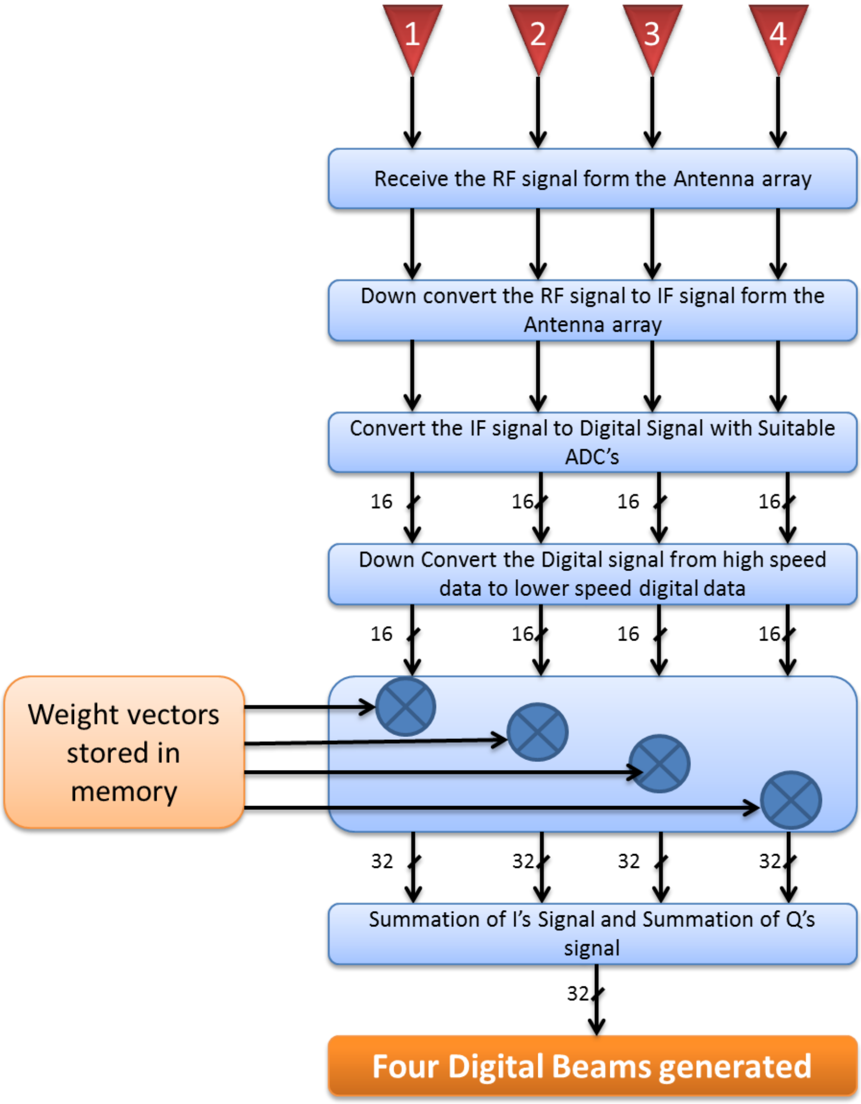

1
2
3
4
Receive the RF signal form the Antenna array
Down convert the RF signal to IF signal form the Antenna array
Convert the IF signal to Digital Signal with Suitable ADC's
16
16
16
16
Down Convert the Digital signal from high speed data to lower speed digital data
16
16
16
16
Weight vectors stored in memory
32
32
32
32
Summation of I's Signal and Summation of Q's signal
32
Four Digital Beams generated

Figure 5-6: Data flow chart of four element fixed weight beam former to form four digital beams.

The antenna elements will receive the RF signal from the space and in the next stage the RF signal is converted to lower intermediate frequency. The suitable digitizers are used to convert the lower intermediate frequency from analog domain to digital domain. The digital signal will be processed further to reduce the data rate to lower speed by using the signal processing technique. The offline weights are calculated using the optimal and fast convergence algorithm and stored in the memory. The stored weight vectors are suitably multiplied with the input signal to form the multiple numbers of beams in the digital domain as shown in the figure below.

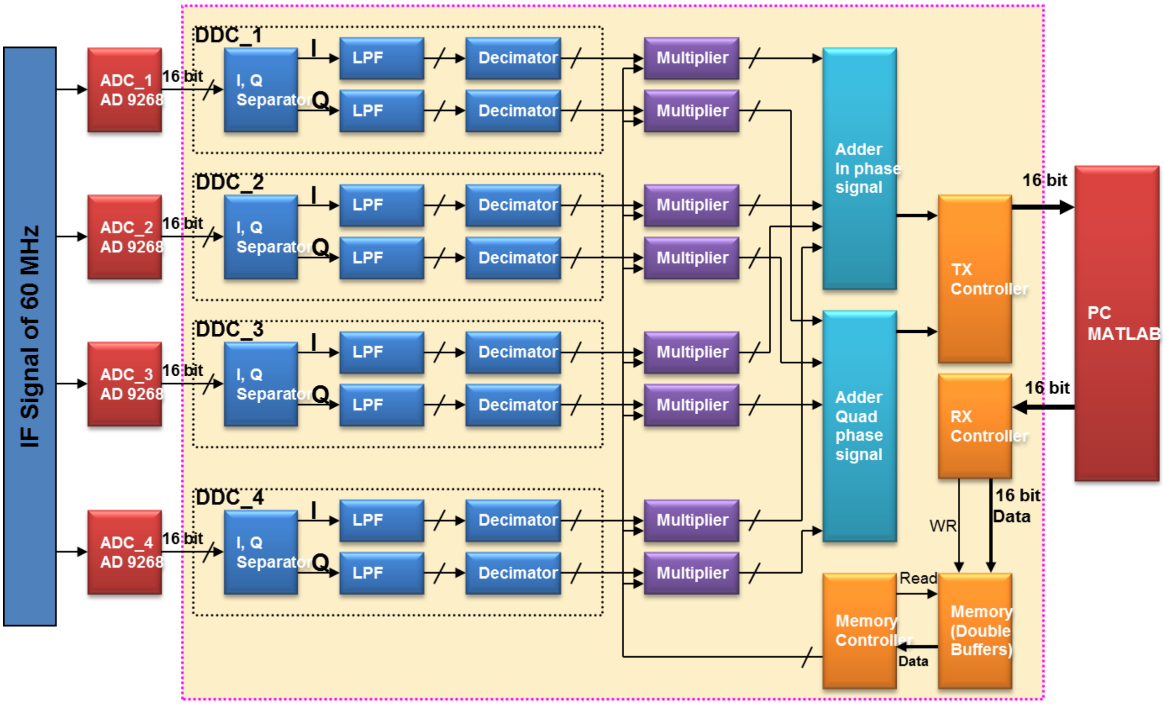
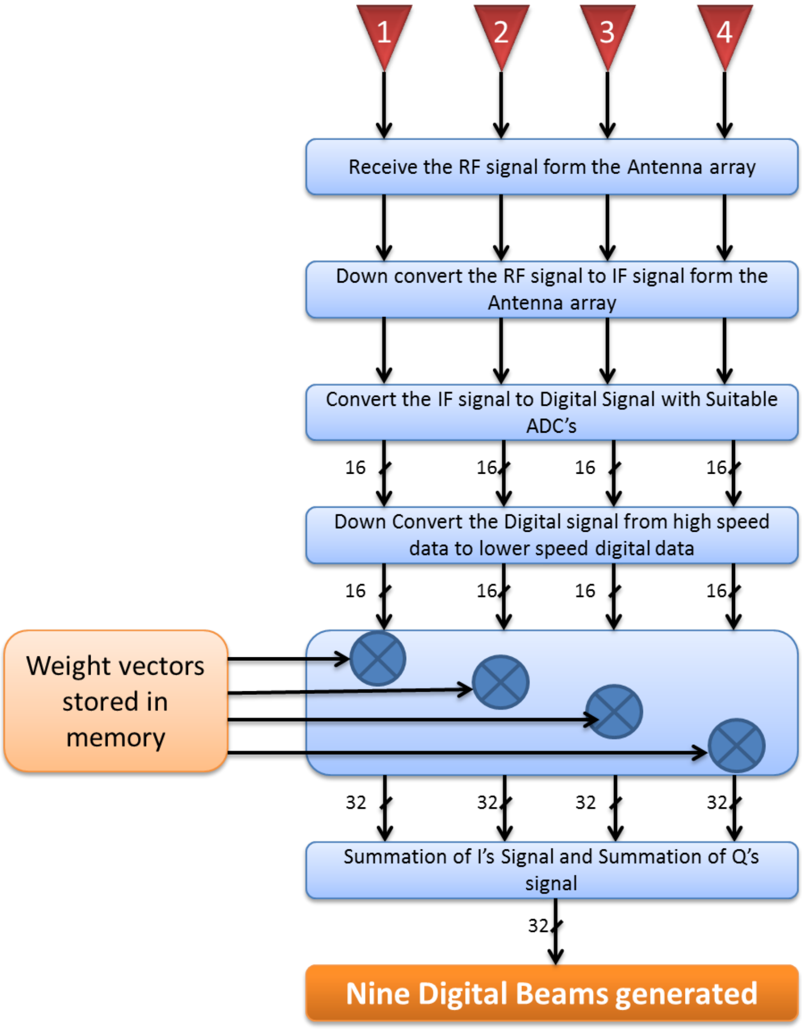


Figure 5-7: Architecture of Four element fixed weight beam former to form nine digital beams.

5.7 Fixed weight Sixteen element Beam Former architecture to form nine simultaneous beams.

An architecture is developed to form nine receive beams in the receive mode of radar operation simultaneously. The data flow is as shown in Figure 5-8.



1
2
3
4
Receive the RF signal form the Antenna array
Down convert the RF signal to IF signal form the Antenna array
Convert the IF signal to Digital Signal with Suitable ADC's
16
16
16
16
Down Convert the Digital signal from high speed data to lower speed digital data
16
16
16
16
Weight vectors stored in memory
32
32
32
32
Summation of I's Signal and Summation of Q's signal
32
Nine Digital Beams generated

Figure 5-8: Data flow chart of four element fixed weight beam former to form nine digital beams.

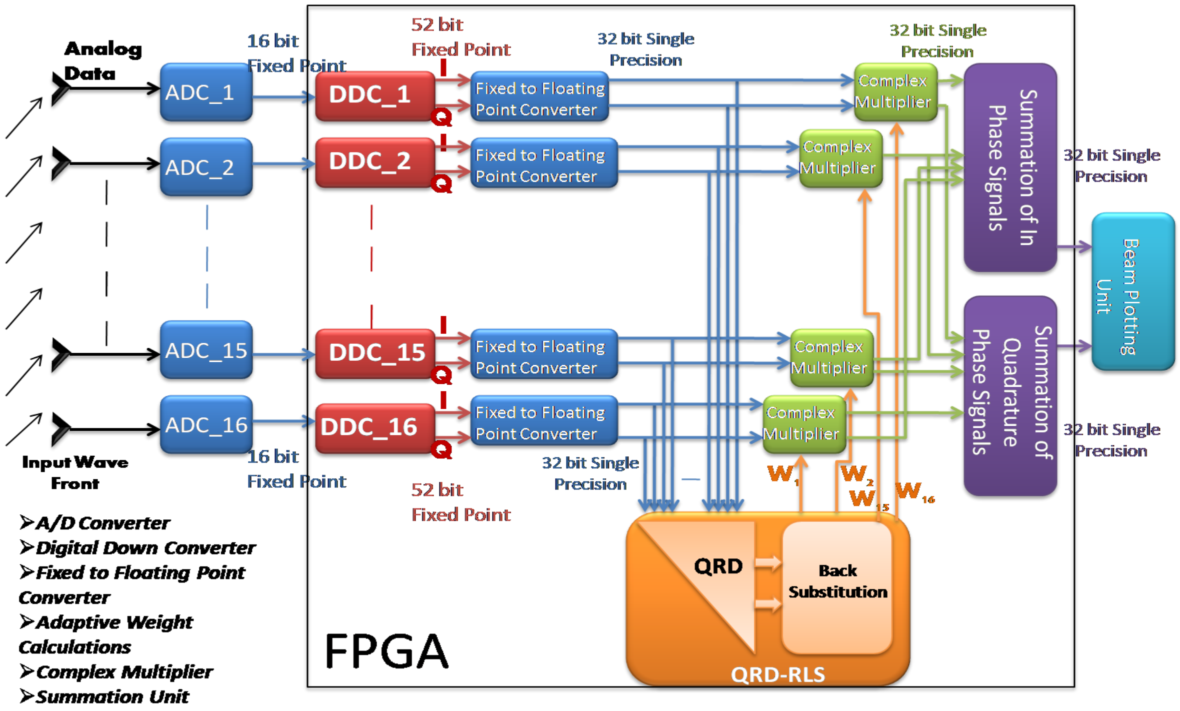


Figure 5-9: Architecture sixteen element fixed weight beam former to form four digital beams.

Out of above mentioned components, the Digital down converter, Complex multipliers, adaptive filter algorithm and summation unit are implemented on the FPGA. The adaptation is achieved by multiplying the signal coming from DDC with complex weights and then summing them together to obtain the desired radiation pattern. These weights are computed adaptively by the signal processor to adapt the pattern to the changes in the signal environment.

5.8 Adaptive beam former architecture to form multiple beams

5.8.1 Data flow of Adaptive beam former architecture

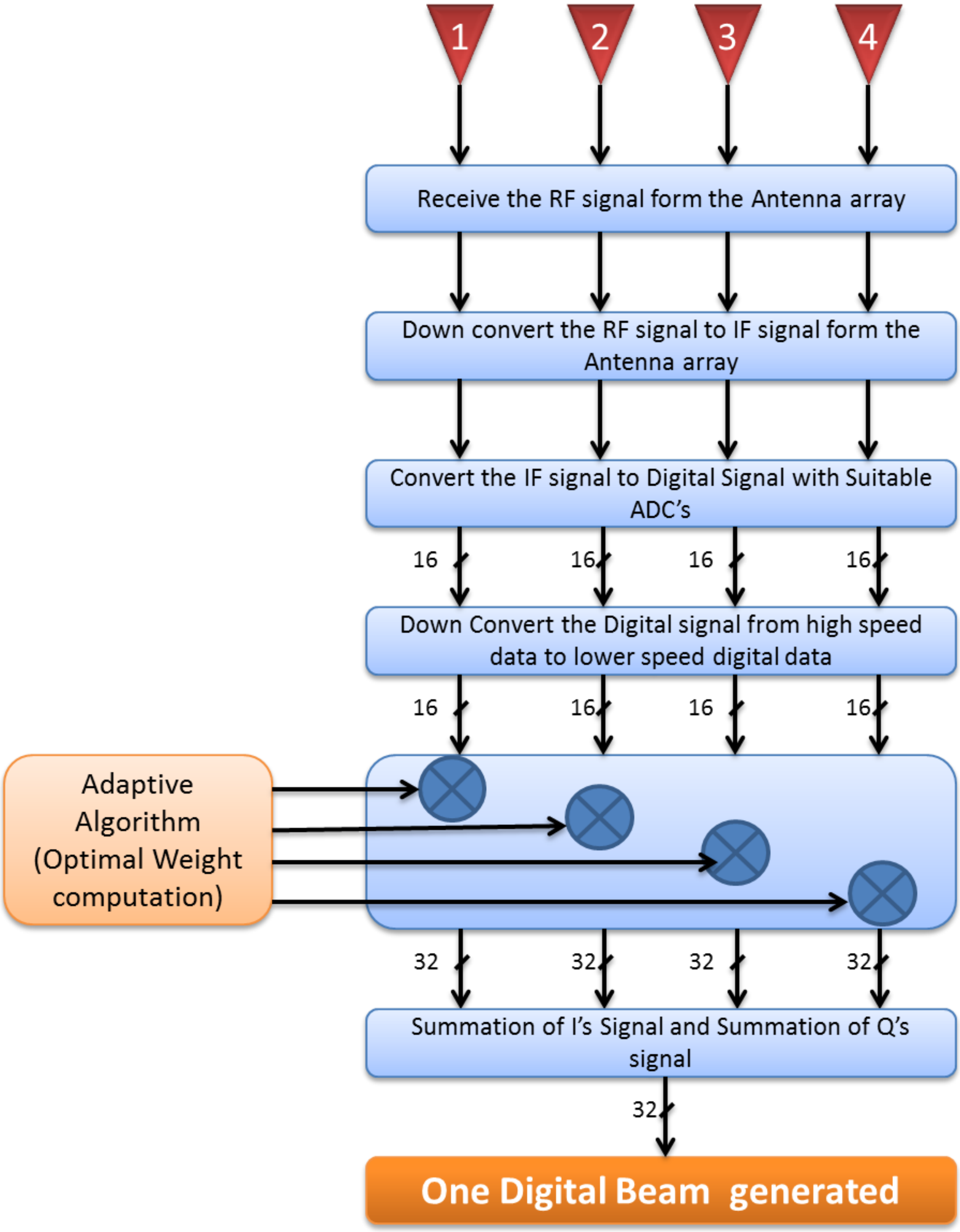

1
2
3
4
Receive the RF signal form the Antenna array
Down convert the RF signal to IF signal form the Antenna array
Convert the IF signal to Digital Signal with Suitable ADC's
16
16
16
16
Down Convert the Digital signal from high speed data to lower speed digital data
16
16
16
16
Adaptive Algorithm (Optimal Weight computation)
32
32
32
32
Summation of I's Signal and Summation of Q's signal
32
One Digital Beam generated

Figure 5-10: Architecture four element fixed weight beam former to form one digital beam.

5.8.2 Fixed point operations based Adaptive beam former architecture

Architecture is developed to form one adaptive digital beam. This consists of a digital down convertor, adaptive weights computation and summation. The data from the ADC will be arriving at 50 mega samples per second (MSPS) this data will be down converted to 5 MSPS by using the suitable down convertor with decimation factor by 10. Further this data will be stored in the FIFO. This data will be used simultaneously for computation of adaptive weights and once the weights are available it will be multiplied with the actual signal to form the beam in that direction. The data arriving from the ADC is 16 bit fixed point format and the same data will be used for down conversion.

The adaptive weight computation is carried out by an adaptive algorithm which will calculate the weight according to the input signal. Then these adaptively calculated weights [17] will be multiplied with the input in phase and quadrature signals with a complex multiplier. Then the complex multiplier outputs will be summed together to get the final beam [41].

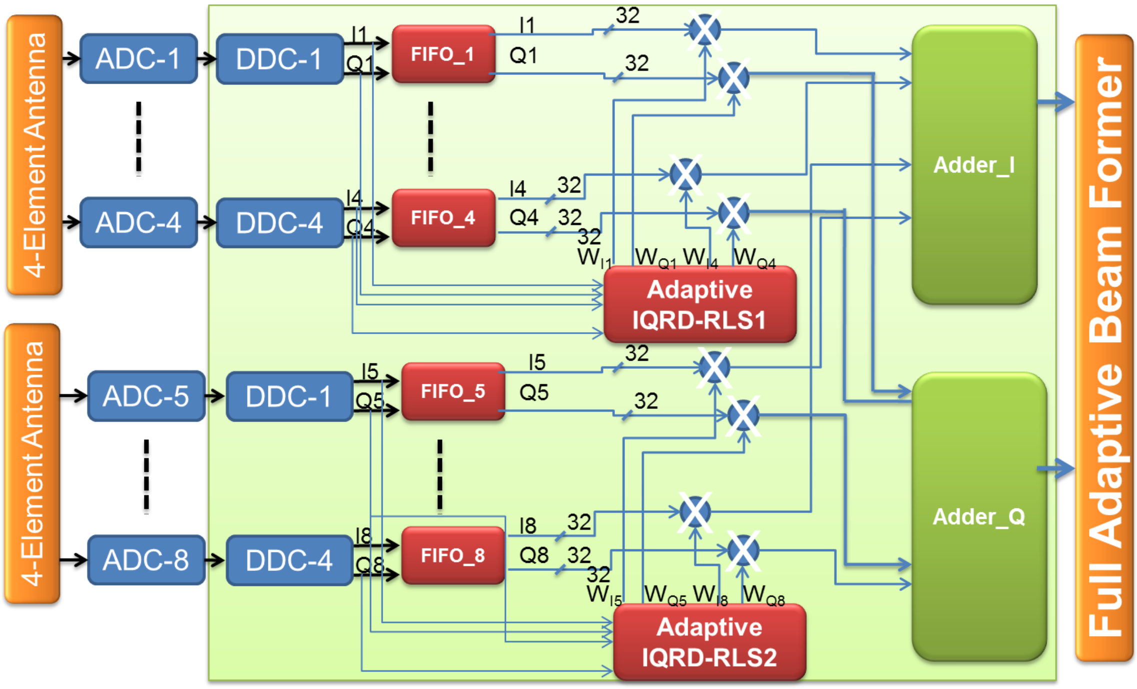


Figure 5-11: VLSI Architecture of FPGA1 and FPGA2 with ADC and DDC

The Figure 5-11 shows the architecture consisting of ADC, DDC, IQRD-RLS weight calculator etc. It take input from the ADC, and then DDC will make two signals from the ADC input by multiplying the ADC output with the NCO(Numerically Controlled Oscillator) output which is sine and cosine waves, producing two frequency translated signals, one in phase with the input signal and the other is quadrature signal. The DDC is used as a frequency translator [47].

The weight computation time for one set of 8 weights supposed to be less than 5 micro seconds. With this architecture it is achieved that the total one adaptive beam formation is 4.6 microseconds. To achieve this architecture designed in parallel processing pipelined manner. All the modules in the architecture will work parallel and data flow will be happening in free flow manner.

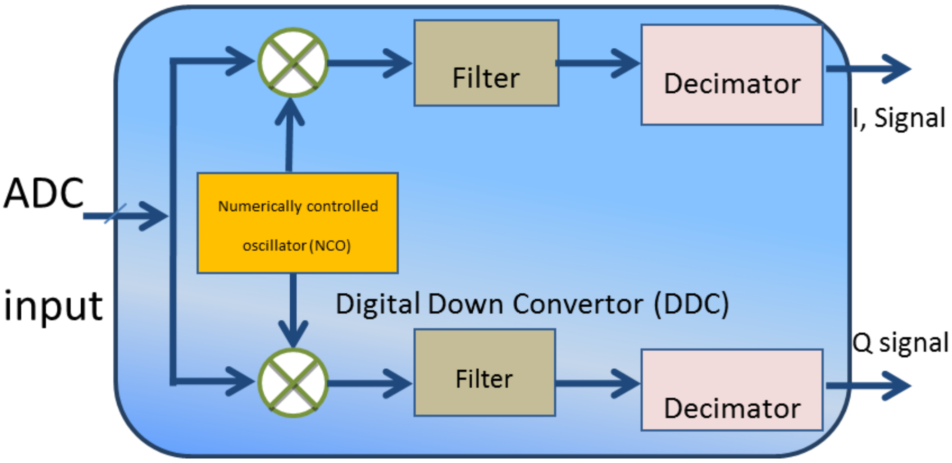

ADC
input
Numerically controlled oscillator (NCO)
Filter
Decimator
I, Signal
Digital Down Convertor (DDC)
Q signal
Filter
Decimator

Figure 5-12: Digital down Convertor (DDC) functional Block Diagram

The ADC is sampling at 50 MSPS with a sampling clock of 50 MHz and the ADC data will be 16 bit which will be received by the mixer. The mixer will also get the reference clock of 50 MHz from the internal clock source. The digital mixer is nothing bu a 14 bit multiplier in time domain where as in frequency domain it translates the higher frequency to lower frequency. To remove unwanted components a suitable low pass filter is used with a band width of 5 MHz. A digital decimator is used to decimate the by a factor of ten, in order to reduce the digital data speed. The output of this module is inphase signal and quadrature phase signal each of 16 bit.

The fixed precision arithmetic representation leads to hardware design that is efficient and also introduces a small amount of error. Here the design employs two's compliment method and the representation of finite precision arithmetic comprise of the sign bit, integer part along with the fractional part. Quantities that find occurrence in the algorithm are characterized with mn bits designed for the integer along with m bits which are employed for the representation of the fractional bits. Finally, representing a finite precision illustration would entail mn+m+1 bit, were one bit is utilized as a sign bit representation [80]. This architecture is most optimized architecture to form one adaptive digital beam in the desired direction within the time of 5 micro seconds.

5.8.3 Floating point operations based Adaptive beam former architecture

The adaptive beam former architecture is designed using the floating point arithmetic.The floating-point arithmetic permits to symbolize broad range of numbers with constant precision. Finite-point arithmetic symbolizes a decreased range of numbers with appropriate precision absolute. Employing floating precision arithmetic is highly expensive when considered in terms of hardware and results in inefficient architecture, particularly when implemented using FPGA.

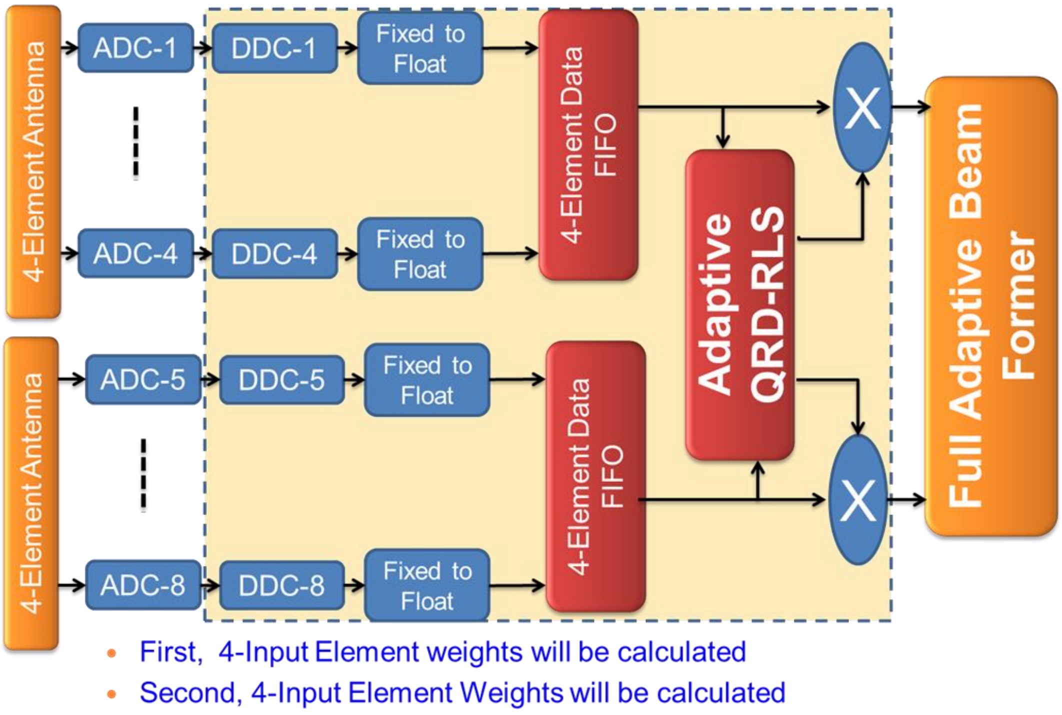


Figure 5-13: Adaptive Beam former VLSI Architecture with Floating Point Operation

QR decomposition needs the usage of arithmetic operations, the calculations required finds an expansion when there is growth in the dimension of the matrix. Large number of calculations is observed during the matrix decomposition process where the upper triangular matrix and the orthogonal matrix are obtained. Calculations employed for the decomposition process are the basic arithmetic computation which is a direct approach; complicated computations significantly alter the precision of the entity likely producing the inefficient values during the implementation in the FPGA. Finite-precision arithmetic reduces the obtained precision and thereby introducing two different types of errors which are coined as the round-off and the truncate error. Round-off error makes its presence when the additional bits are required in the outcome than the bits that are reserved usually after the arithmetic computations. Due to the restricted range of bits that are required to produce numbers, truncation error occurs. These problems must be handed rigorously in order to stop the overflow which results in the erroneous outcome [86].

This architecture employing floating point arithmetic to form one adaptive digital beam is developed and simulations data processing has been carried out. This architecture produces very less error in the bit handling as there is no truncation and rounding off situations. It can handle large number of bits as per the IEEE standard format. The major limitation of this architecture is FPGA resource utilization is huge in numbers. The DSP slices about 41% has been utilized to the adaptive weight calculation using Inverse QR Decomposition algorithm, for a four element array. Since within the available limit of the resources it is not possible to generate the weights simultaneously for 8 elements, the architecture is divided into two parts such that a common IQRD RLS module will be used to generate the first four element weights and in the second part of another 4 element weights are computed. Finally, when all the 8 element weights are available the adaptive digital beam will be formed in the desired direction.

5.9 Summary

In this chapter development of adaptive digital beam former architecture is discussed in detail. The FPGA implementation of systolic array consisting of sub modules like angle processor, rotation processor and weight processor are explained along with the detailed internal design. In conventional QRD RLS, the optimal weight vectors arrive in two steps viz; QR decomposition and back substitution. This brings the major limitation in achieving the processing time for the phased array application. The Inverse QRD RLS method brings the advantage over the conventional method and process the matrix inversion simultaneously and generating the optimal weight vectors. The fixed weight beam former architecture is discussed along with benefits and limitations. The adaptive weight calculation for sixteen element planar phased array architecture is explained and the data flow is covered starting from the antenna element level digitization to multiple beam formation. This architecture is developed for floating point operations as well as fixed point operations. The comparison has been brought out listing the benefits and limitations.

In the next chapter hardware realizations using the multiple FPGAs were covered. The multiple beam former architecture is developed initially with Virtex-V FPGA considering the four element array and the weights were calculated offline and stored in the memory of this board. The architecture is extended further to a sixteen element planar array and the optimal weights were calculated online and multiple beams were formed using the Virtex-VI FPGA configuration board. The optimized architecture was not meeting the processing time the architecture was split into three FPGAs and the architecture is redesigned to form multiple adaptive digital beams using fixed point arithmetic and floating point arithmetic operations. The resource utilization for all the architectures is compared and is tabulated.

Chapter 6

Hardware Realization of Adaptive Beam Former Architecture

Chapter 6.

Hardware Realization of Adaptive Beam Former Architecture

6.1 Realization of Fixed Weight Digital Beam Former

The digital beam former architecture is realized by considering the following constraints.

- The total time for weight computation and beam formation should be less than five micro seconds.
- The resources should be optimized and modular in nature so that the developed architecture can be extended to larger dimension of the array size.

The fixed weight beam former architecture is implemented on Virtex-V FPGA. The architecture is shown in the Figure 6.1.

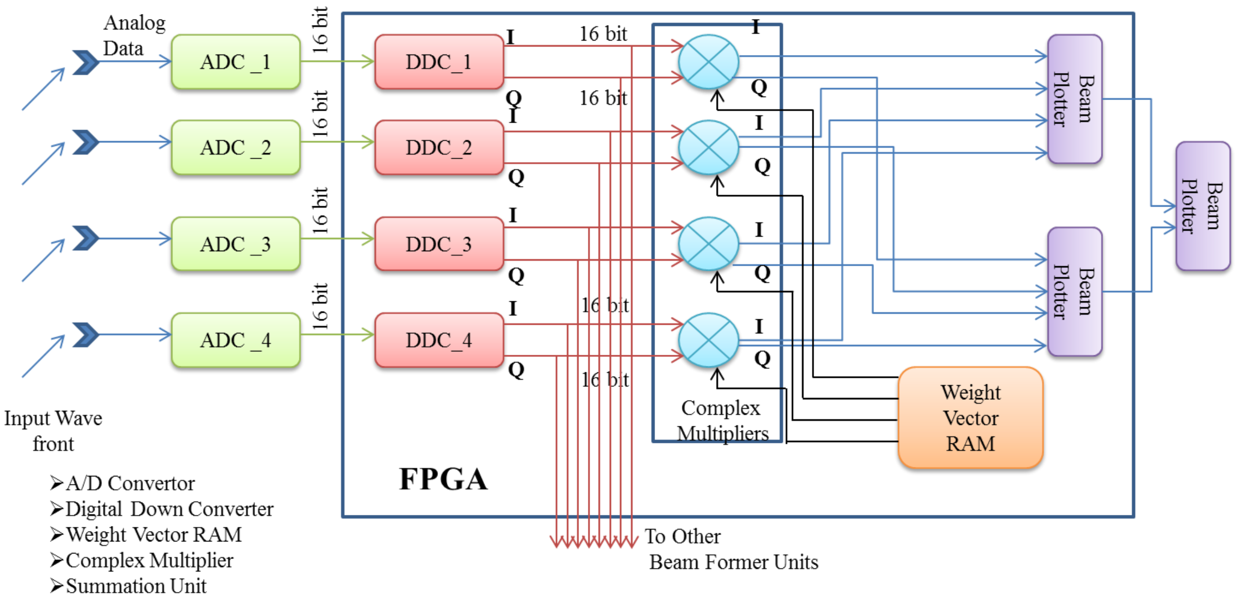


Figure 6-1: Fixed weight DBF architecture for four element array

The digital beam former architecture shown in Figure. 6-2 describe the four element phased array antenna configuration. This Architecture is generic in nature and can be extended to any number of larger array size. The basic modules of this architecture are Digital Down Converters (DDC), complex Multipliers and complex adders. The IF signal, generally is in the range of 50 MHz to 60 MHz is converted equivalent to digital data using 8/16 bit format, 125 MSPS high speed ADCs. As per the Nyquist criteria it is band pass sampling as the signal bandwidth is 5 MHz, sampling is carried out at 50 MHz and then processed.

The mixer will get the reference clock of 50 MHz from the internal clock source. The digital mixer is nothing but a 14 bit multiplier in time domain where as in frequency domain it translates the higher frequency to lower frequency. To remove unwanted components a suitable low pass filter is used with a band width of 5 MHz. A digital decimator is used to decimate by a factor of ten, in order to reduce the digital data speed. The output of this module is In-phase signal and Quadrature phase signal each of 16 bits.

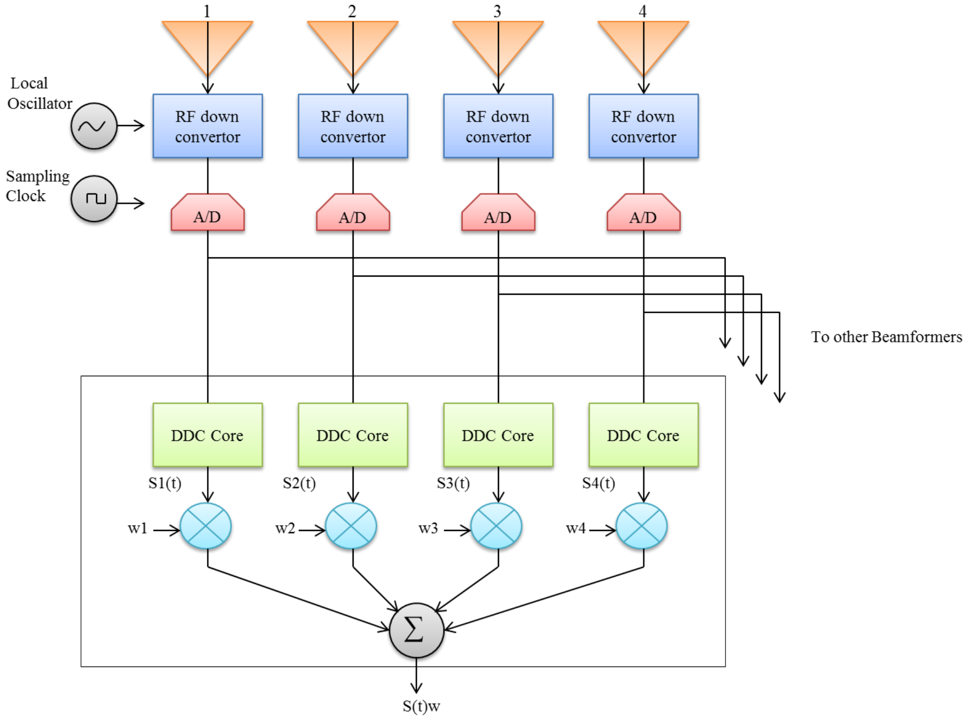


Figure 6-2: DBF architecture for four element array.

A high speed ADC is used with a resolution of 16-bit; this data is given to FPGA. The In -phase and Quadrature phase signals are generated using a custom made DDC core as shown in Figure 6-3.

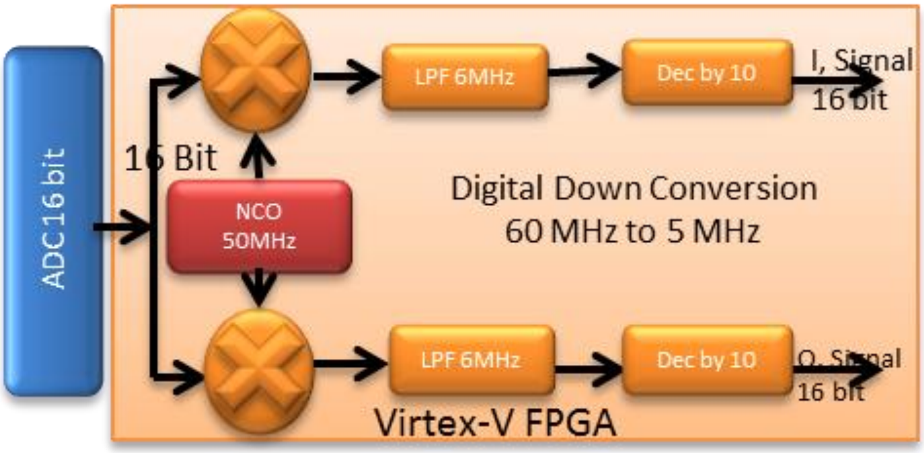


Figure 6-3: Digital down convertor internal architecture

Following Figure 6-4 shows, Digital down convertor implementation, which requires two multipliers, one each for the sine and the cosine.

Input data are multiplied by the Quadrature sine and cosine waveforms, to achieve a frequency translation to the base band as shown in Figure. 6-5.

It can be observed that base band signal will be moved to zero frequency from centered at 60 MHz after the decimation by 10 using the above formulae as well as the Nyquist zones shown. The modules used are 16 x 16 bit signed multipliers. As there is a growth of bits during the multiplication, the higher 16-bits of the total output is selected and lower LSB are deleted for subsequent processing as it contains mainly noise. The quantization error is acceptable limit of 0.1%. This processing is for one element or one channel at a time. So to consider for sixteen elements array then the similar 16 channel architecture has to be used. Each one of the channel will process independently. The complete architecture for a four element array is shown in Figure 6-6. The modules developed in VHDL are complex multipliers and complex adders. To perform this complex multiplication in FPGA it is required to carry out equivalent floating point arithmetic operations in fixed point arithmetic.

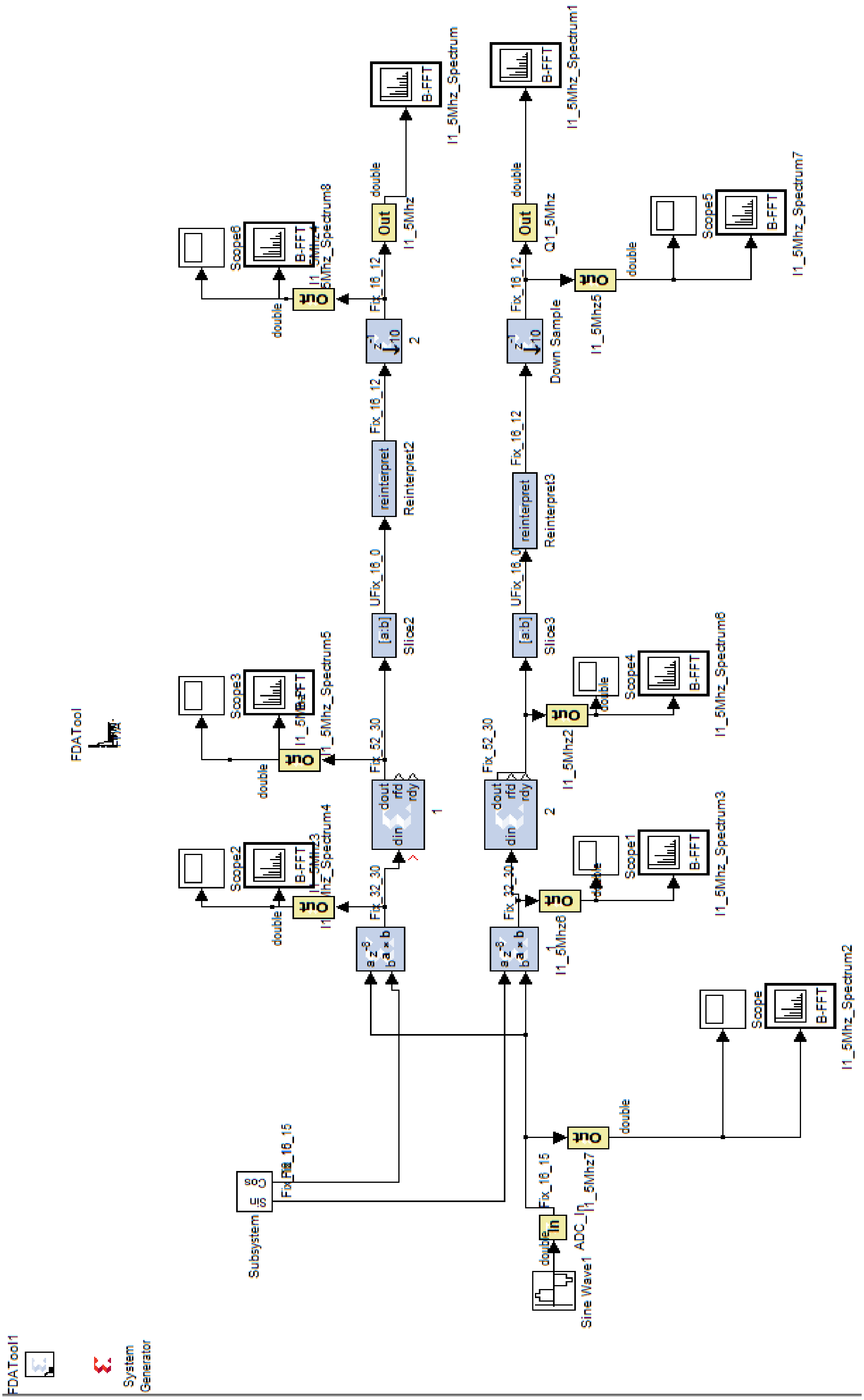


Figure 6-4: Digital down convertor implementation.

$$\mathit{if}\ \mathit{fix}\left(\frac{f_c}{\frac{F_s}{2}}\right)\ \mathit{is}\begin{Bmatrix} \mathit{even}, f_{IF=rem(f_C,F_5)} \\ \mathit{odd}, f_{IF=f_s-rem(f_C,F_5)} \end{Bmatrix}$$

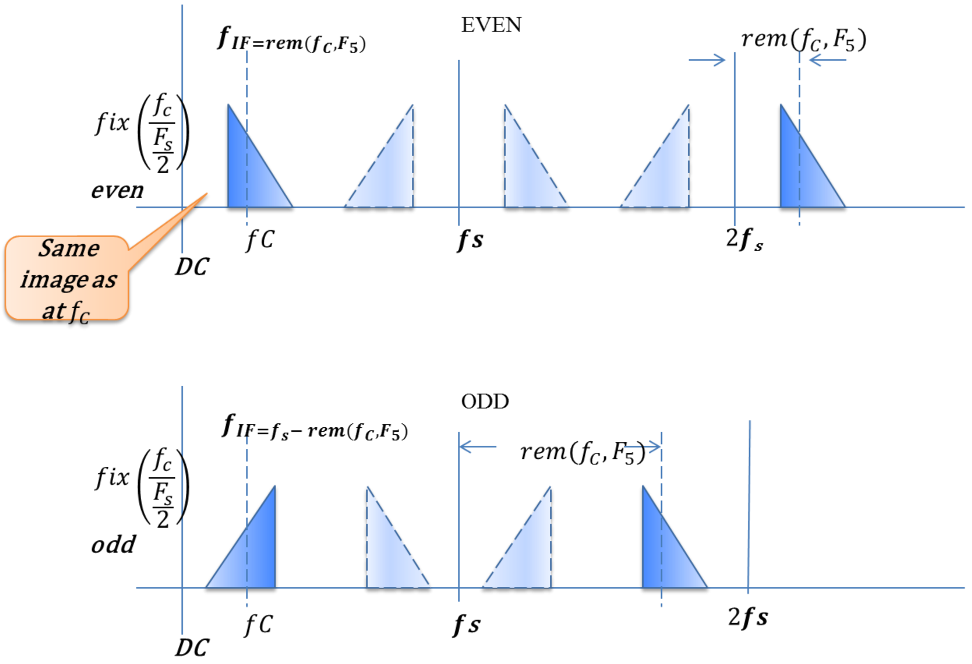


Figure 6-5: Nyquist zone for fc=60 MHz and fs=50MHz

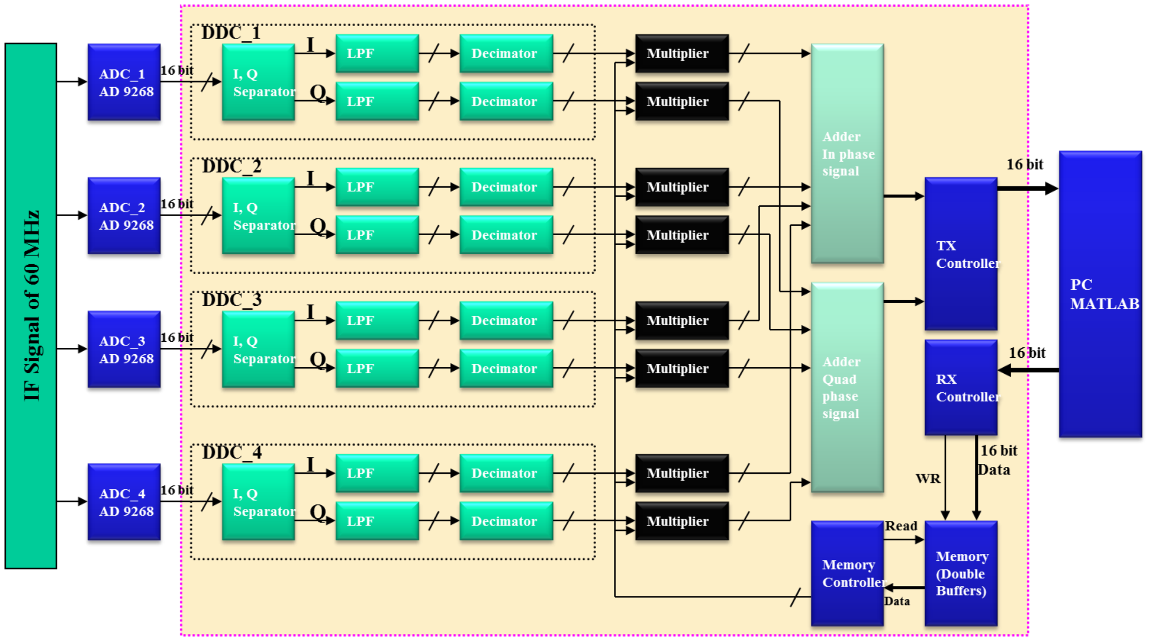


Figure 6-6: DBF architecture for four element array

Multiple beams are formed based on the optimal weights or steering vector. Weights are computed offline i.e in the predetermined direction with respect direction of arrival and stored in Block RAM of FPGA. With respect to the direction of arrival within the span of -

50 deg to + 50 deg, weight vectors will be applied and as many number of multiple beams can be formed. It is assumed that the direction of arrival is known before hand as the Radar computer is scheduling the transmit beam. Multiple beams are generated with respect to the direction of arrival. The offset is decided based on the weights computed and deposited in the in the Block RAM of FPGA. This architecture is designed and weights are calculated for +/-10 deg, +/-20 deg and so on. The weights will decide the multiple numbers of beams in the direction of arrival and complex multiplication is required to compute to form multiple beams. For an N element array to form one digital beam N weights are required and to form two beams out of N elements 2xN weights are required. It is assumed that the weights are calculated apriori for required number of beams and stored in the memory.

The data flow from the element to ADC to and further down conversion is shown in Figure 6-6. The functional simulation is carried out in VHDL and realized on the prototype hardware shown in Figure. 6-7. For an N-element array the beam will be formed based on the summation of all the partial beams in the same digital domain, B (t), given by equation 6.1.

$$B(t) = \sum_{k=0}^{N} S_k(t) * W_k \qquad (6\text{-}1)$$

To form multiple receive beams, the results are stored in memory for processing.

where,

N : Number of Elements in the phased array

W_k : Complex Weight of K_{th} Element

$S_k(t)$: Signal Received from the antenna array.

The following development hardware shown in Figure 6-7 is used to implement digital beam former architecture to form multiple beams.

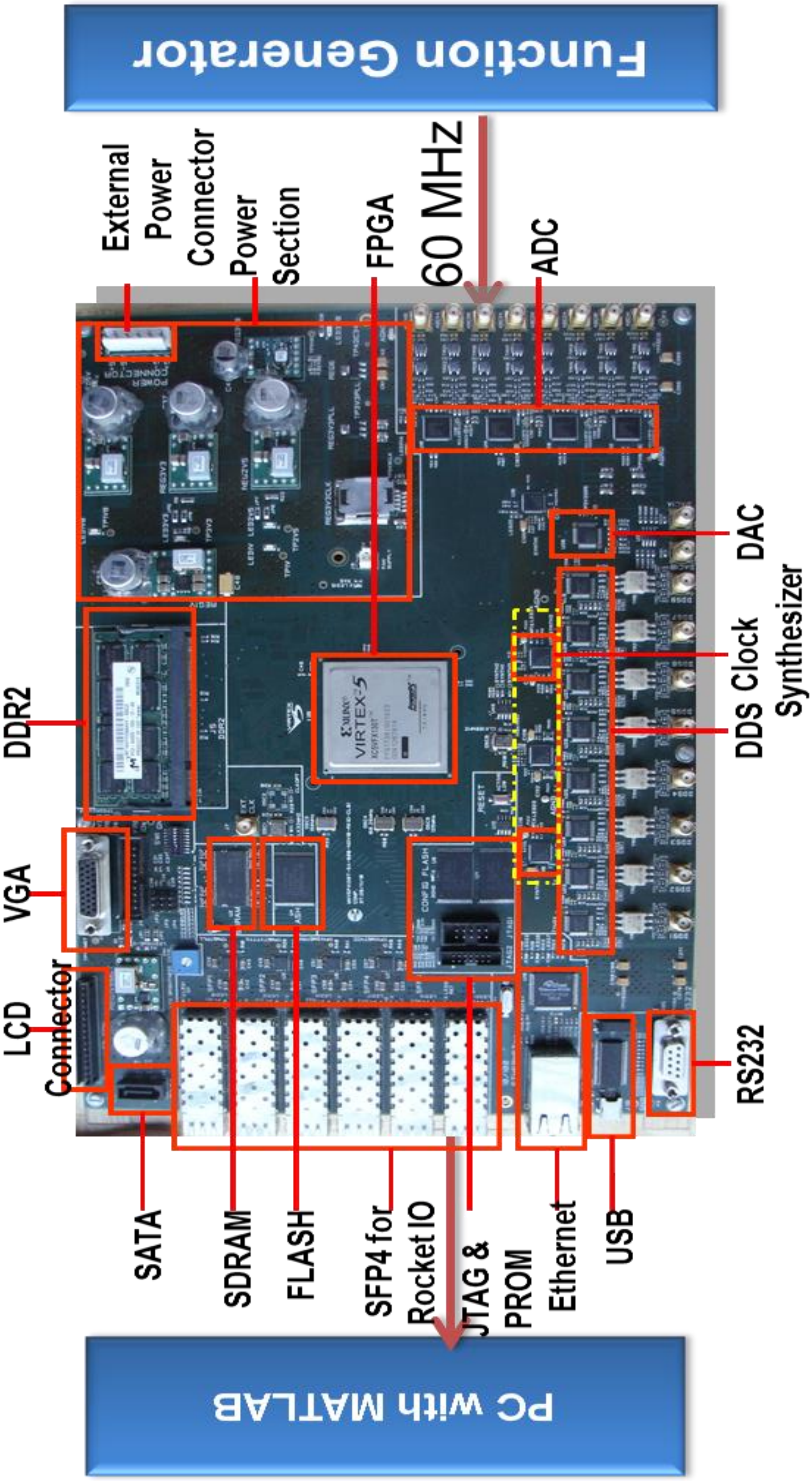


Figure 6-7: Digital beam former architecture developed using prototype hardware.

The architecture developed is modular in nature and can be converted to design an ASIC as large numbers of chips are required for the Digital phased array Radar development.

The proto hardware consists of following important features.

- FPGA on board is Virtex 5 FX130T
- Multiple Clock Domains used are
  - 156.25 MHz clock oscillator for SFP for external communication
  - 150 MHz clock oscillator for SATA interface
  - Onboard Clock Oscillators : 32 MHz
- Multiple ADC and DAC are used also they are synchronized.
- For storing the data a Two GB DDR2 – SDRAM
- Flash Memory on board is 256Mb and 128 Mb SDRAM Memories
- Rocket IO interface @ 3.125 Gb/s
- Six SFP connectors are provided for SFP modules
- Analog Input is given to AD 9268 a dual channel ADC 16 bit and 125 MSPS and DAC, DAC-2904 14 bit and 125 MSPS.
- Interfaces for the external world are Ethernet, USB 2.0 High Speed, Two RS-232 channel using MAX3223 on DB9, LVDS Interface.

6.2 Realization of Beam Former Using Systolic Array

The systolic array method is followed to generate the optimal weights. The architecture Block diagram is shown in Figure 6-8, describes the FPGA architectural features of the Digital Beam Former for 16/8 element Phased Array antenna [10].

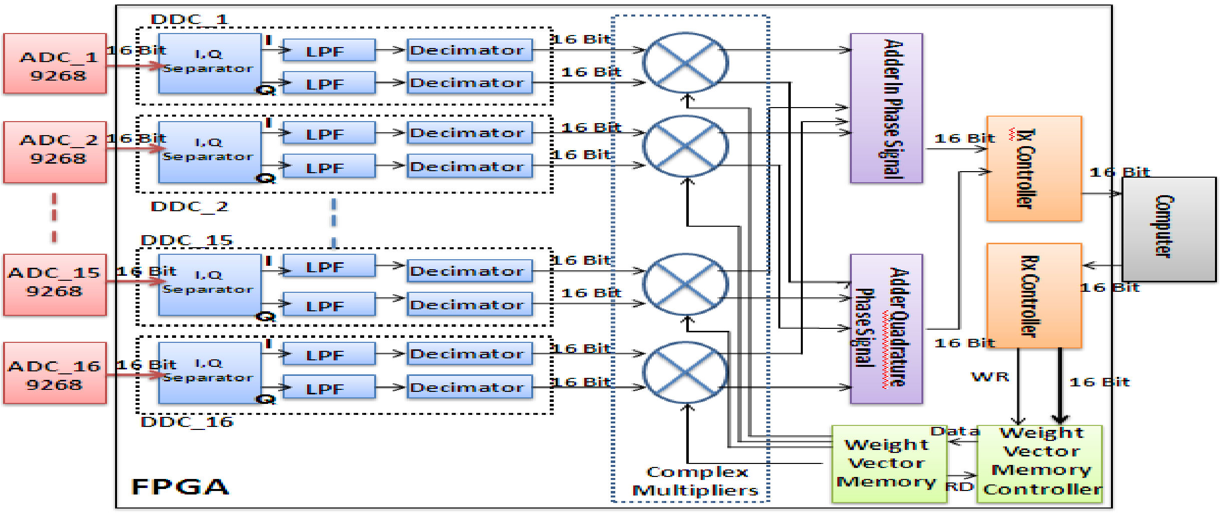


Figure 6-8: sixteen element DBF architecture for generation of multiple digital beams.

This Architecture has been extended from the previous work of four element array to sixteen element array. The architecture designed is modular and generic in nature, further it can be extended to large dimension of the phased array. The basic building block for this architecture remains the same as per previous architecture consisting sixteen modules of Digital down converters (DDC), sampling at the rate of 50 MSPS and down converting to 5 MSPS in the form of 16 bit In-phase signal and 16 bit Quadrature phase signal as shown in Figure 6-9.

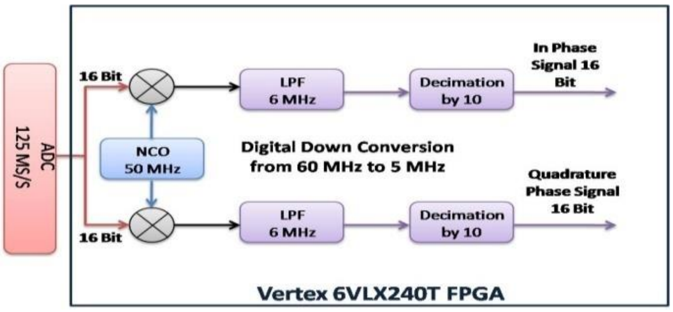


Figure 6-9: FPGA based digital down convertor

The simulation and modeling of one digital down convertor module is shown in Figure 6-10 indicating the Low Pass filter I and Q separator and suitable decimator.

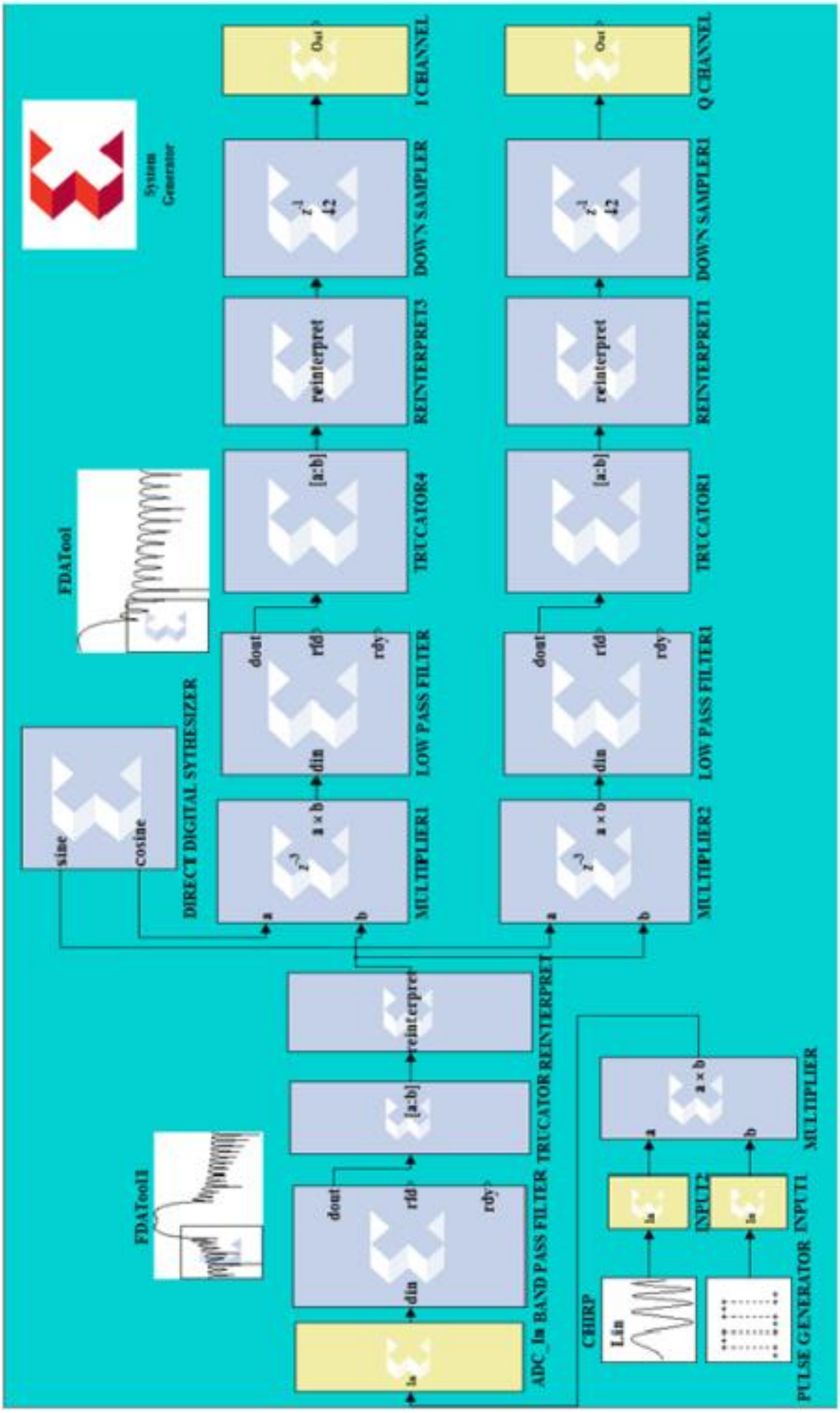


Figure 6-10: Digital down convertor implementation.

The input data sequence is multiplied with the sine and cosine waveforms. This in frequency domain higher frequency translates to lower frequency as shown in Figure 6-11.

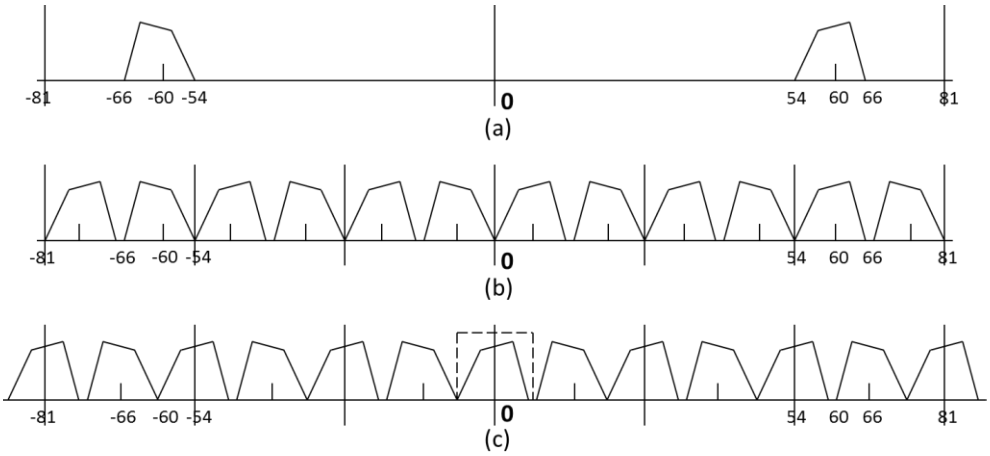


Figure 6-11:Nyquist zones for f_c=60MHz and f_s=50MHz

A VHDL [11] source code is developed to realize the architecture consisting of complex multipliers and complex adders in the hardware. The data available from the ADC’s are in the fixed point format of 16 bit. Further in the array signal processing complex multiplications were carried out and this results into a large amount of bit growth from 16 bit to 32 bit to 64 bit and so on. To curb this bit growth fixed point is represented in the floating point and processing is carried out inside the FPGA.

This sixteen element architecture is developed using the proto hardware consisting of VIRTEX-VI FPGA as shown in Figure 6-12.

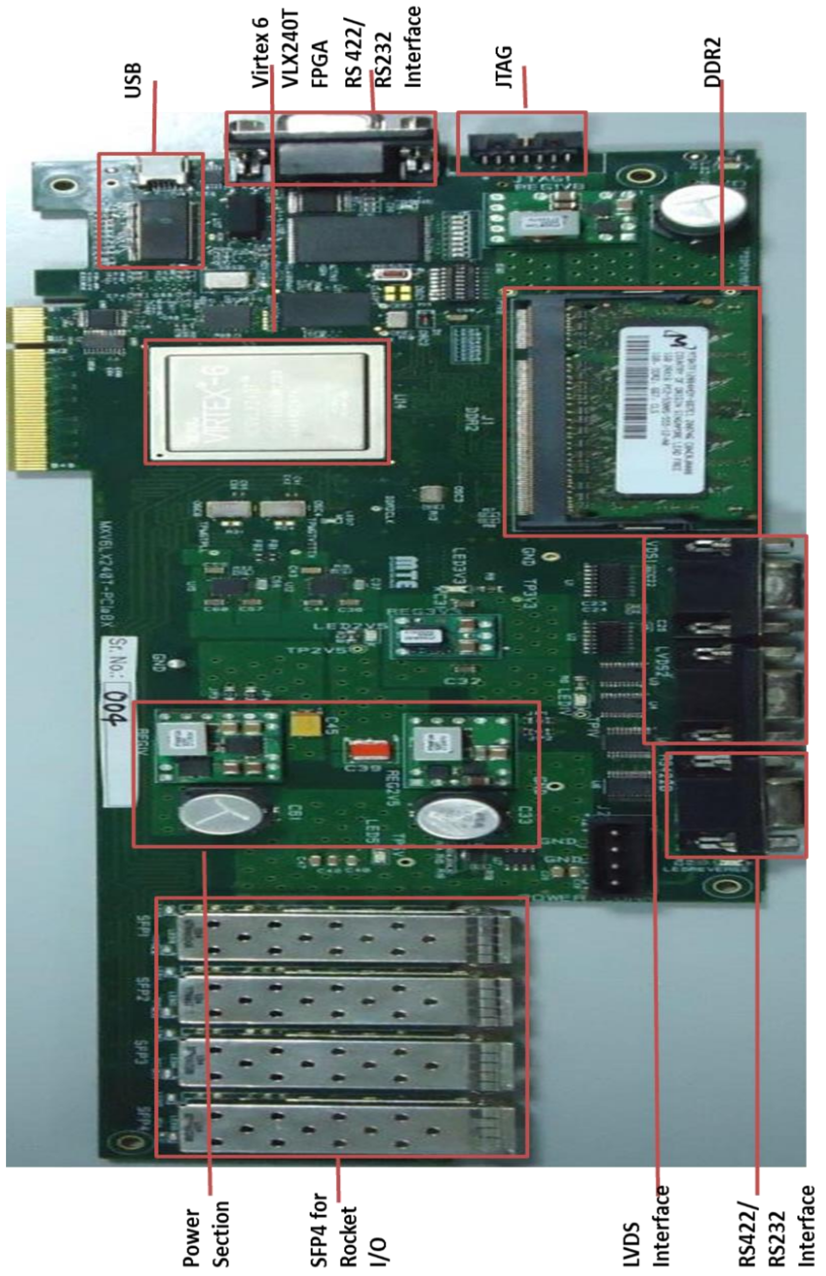


Figure 6-12: Prototype hardware used to form nine digital beams in the direction of arrival.

The Main features of the proto hardware as mentioned below:

- FPGA- Virtex - 6 LX240T – 1 FF1156C [6].
- Multiple Clocks generated using Oscillators: 32 MHz and 156.25 MHz
- Memory on board: ONE GB DDR2, and 256 Mb flash Memory
- High speed serial interface 3.125 Gbps consisting of Four SFP Modules.
- PCI interface : 8x lane @ 2.5 Gbps
- High speed USB 2.0 for external communication.

6.3 Realization of adaptive beam former architecture

The previous two architecture developed are having limitations where in the beams are formed for the sixteen elements but in the direction of arrival as the weights are fixed in nature. The adaptive weight computation is possible using multi FPGA configuration as shown in Figure 6-13.

The architecture of the Array Signal Processor is divided into three different FPGA's for the purpose of the modular design approach. First and second FPGAs are doing almost same work, taking input from the ADC's, frequency translation of signal using DDCs, weight calculating adaptively, followed by multiplying the input signal with the weights. These multiplier outputs are fed to the third FPGA which performs the summation of these signals from both FPGA1 and FPGA2, and produce the beam.

Eight inputs are connected to FPGA1and another eight inputs connected to FPGA2. The net result is sixteen channel inputs; both are synchronously running under the control of FPGA3. The Figure 6-14 is the architecture with ADC, DDC, IQRD-RLS weight calculator etc. It take input from the ADC, and then DDC will make two signals from the ADC input by multiplying the ADC output with the NCO (Numerically Controlled Oscillator) output which is sine and cosine waves, producing two frequency translated signals, one in phase with the input signal and the other is quadrature signal. The DDC is used as a frequency translator [47].

It is possible to make another structure of the FPGA1 and FPGA2 such that a DDS is designed inside the FPGA so that the DDC module can be avoided, in turn, resulting the reception of signals from the ADC modules are not needed, which is best method for the testing environment. The frequency, phase, and amplitude of the DDS output signal can be changed as per the requirement. The DDS module will generate sine and cosine signals in IF frequency or baseband region, upon our commands. For that, in FPGA3, RS232 interface is available such that a command can be given through PC and then FPGA3 will give these signals to both FPGA1 and FPGA2, which is very useful for the testing of the Array Signal processor, as in Figure 6-15. Such hardware architecture of FPGA1, FPGA2 and FPGA3 is shown in Figure 6-16. In this case, architecture of the FPGA3 will be giving the control signals to the FPGA1 and FPGA2 about the Frequency, Phase and Amplitude of the signal to be made by the DDS. FPGA3 will do the final summing of the complex multiplier outputs from FPGA1 and FPGA2, as shown in Figure. 6-14.

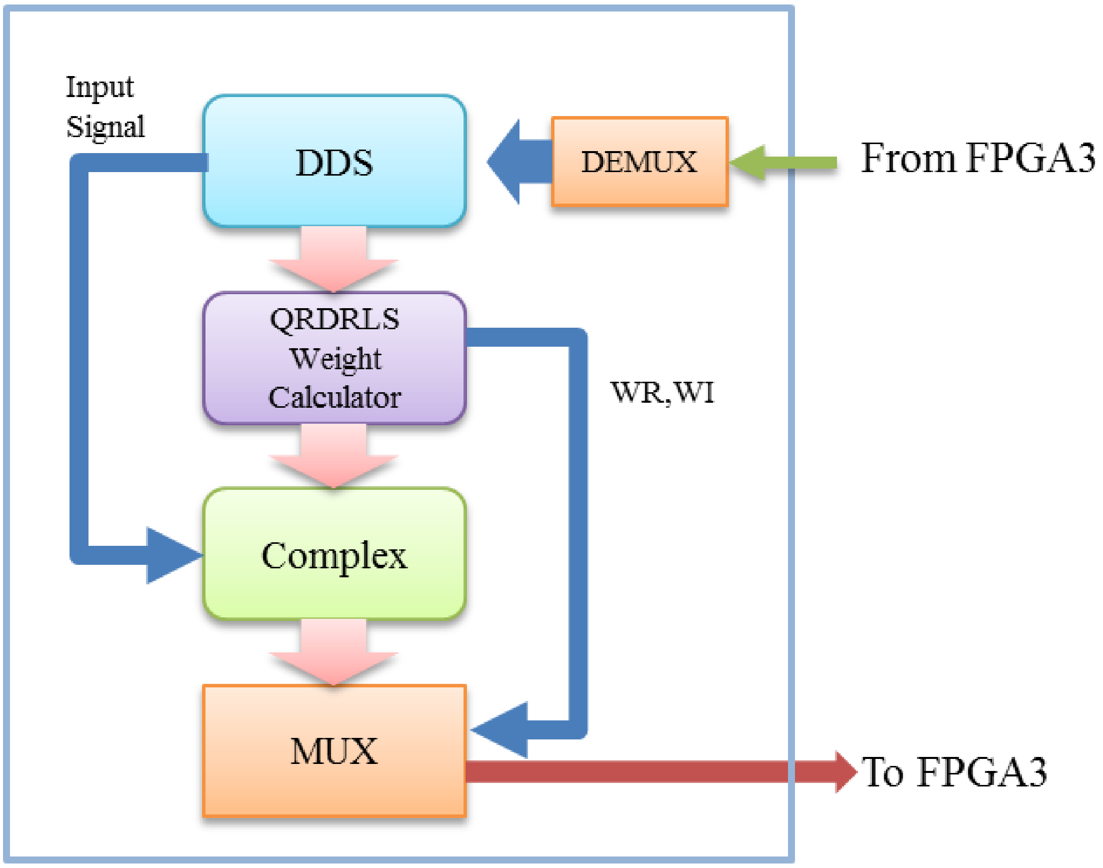


Figure 6-13: VLSI Architecture of FPGA1 and FPGA2 without ADC and DDC

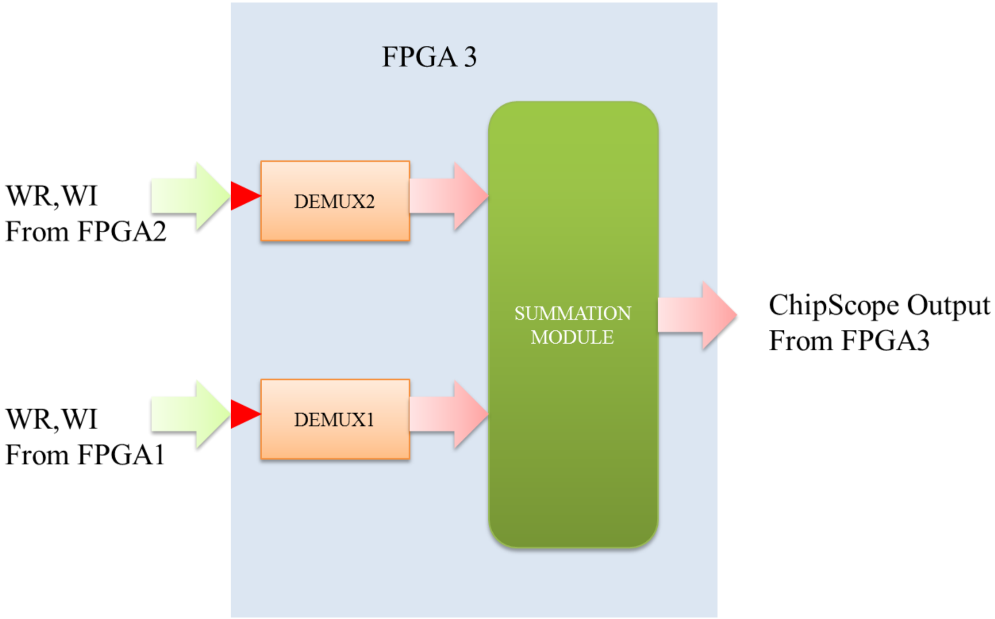


Figure 6-14: VLSI Architecture of FPGA3

When the FPGA1 and FPGA2 outputs are available, it will give output available signal to the FPGA3. On receiving the output available signal from FPGA1 and 2, FPGA3 will initiate dataflow through the data bus synchronously. This implemented architecture is for eight element array. The architecture developed is modular in nature and can be scaled up to larger dimension of phased array Radars.

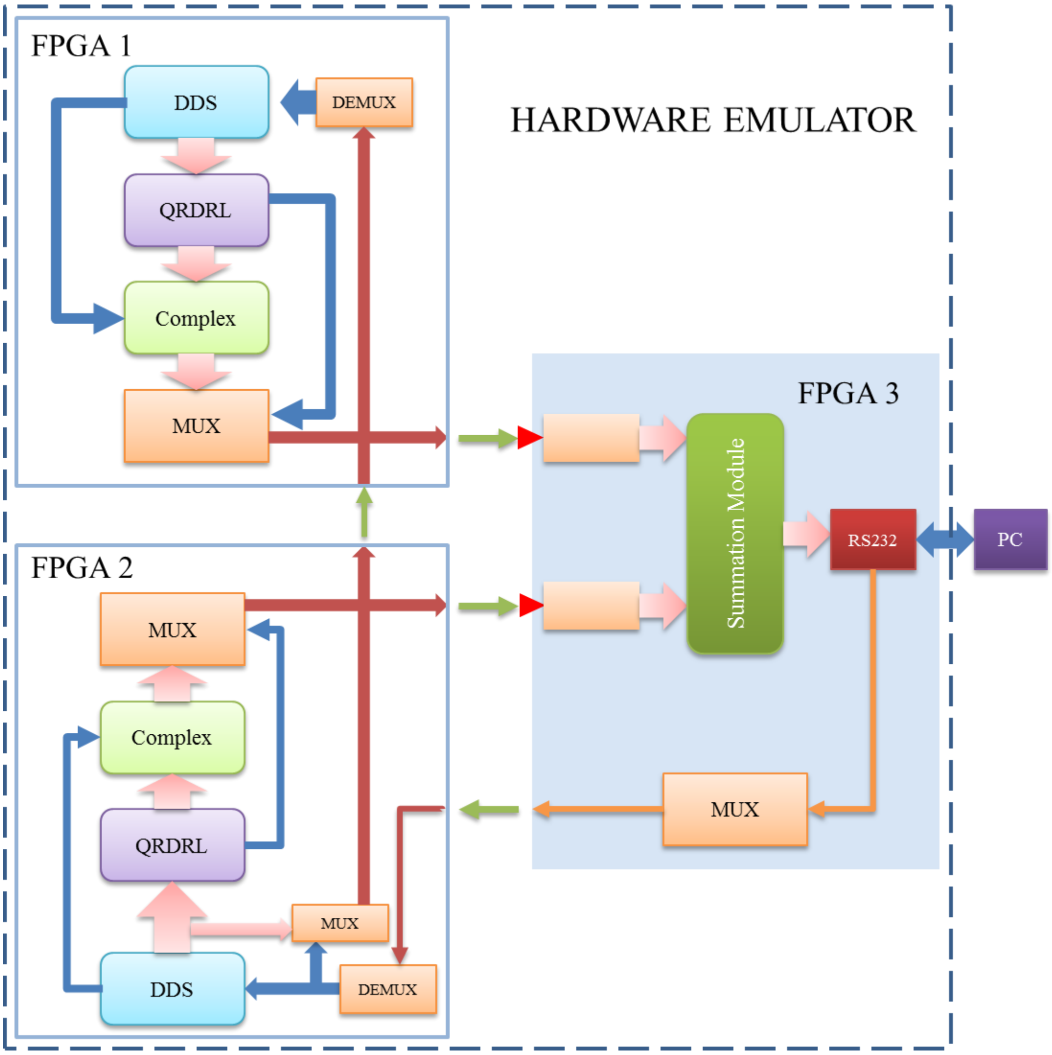


Figure 6-15: Complete test setup including Hardware Emulator

The Complex multipliers and IQRD-RLS weight calculator are 32 bit floating point arithmetic giving better resolution. So, a Fixed to Float converter [16] is used in the DDC output and to convert the fixed point format to the 32 bit binary floating format (IEEE-754 format).

The hardware used for the Implementation of the Adaptive Beam Former VLSI Architecture is shown in Figure 6.17.

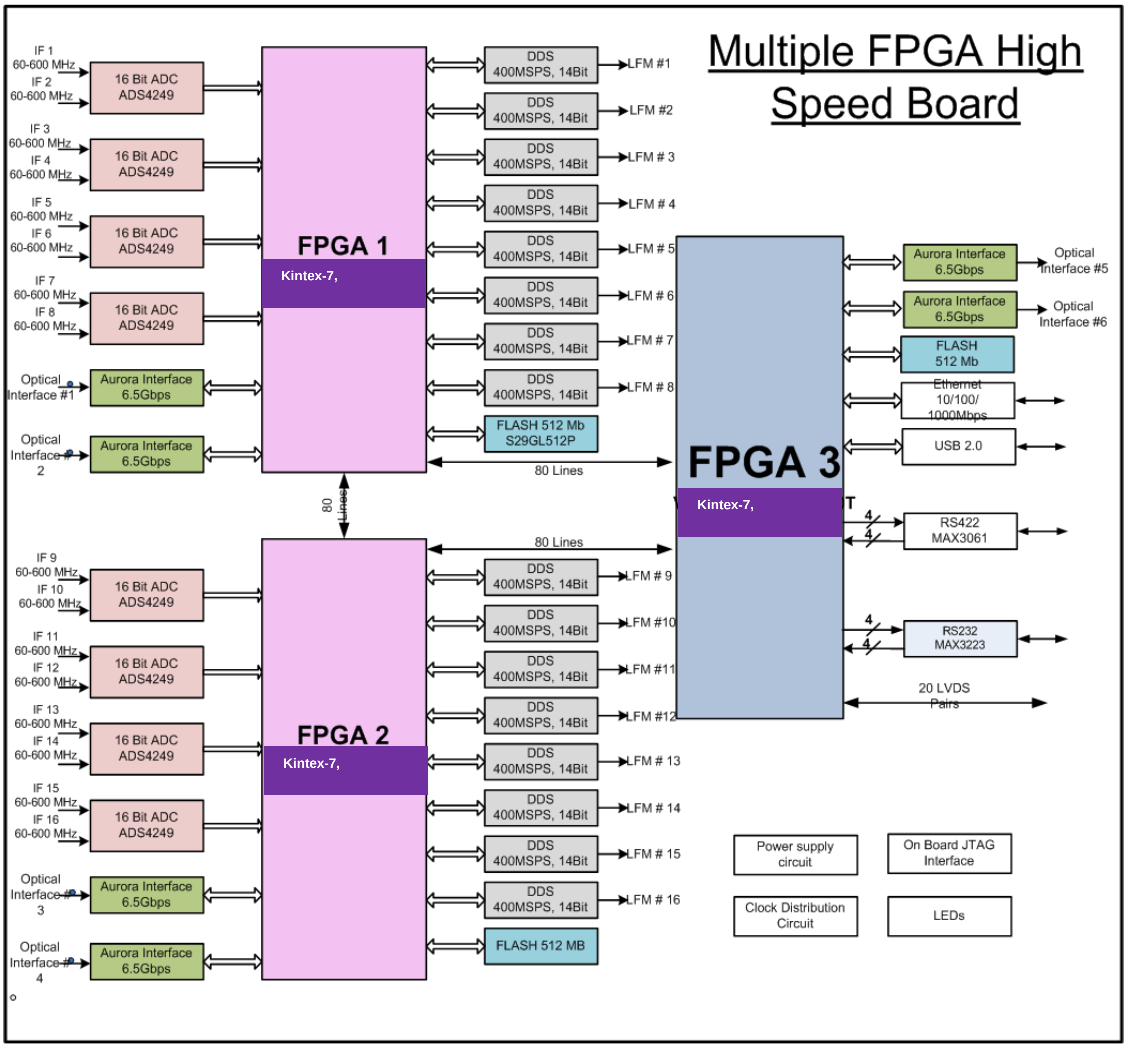


Figure 6-16: Block diagram of Hardware for Adaptive Beam Former VLSI Architecture

The important features of the hardware emulator are below:

- FPGA- Kintex-7 XC7K410T– 1FBG900 [2].
- Clock: Onboard Oscillators: 100 MHz and 200 MHz
- Memory: One Flash 64M x 16 Memory – S29GL512S from SPANSION
- Optical interface @ 6.5 Gb/s
- Dual channel 14 bit ADC, ADS4249.

The hardware used for the implementation of Architecture is shown in Figure 6-18. The tools used for the implementation of Array Signal Processor are

- MATLAB 2010a for algorithm modeling and simulation purpose.
- Xilinx ISE 13.4 for functional simulation and implementation.

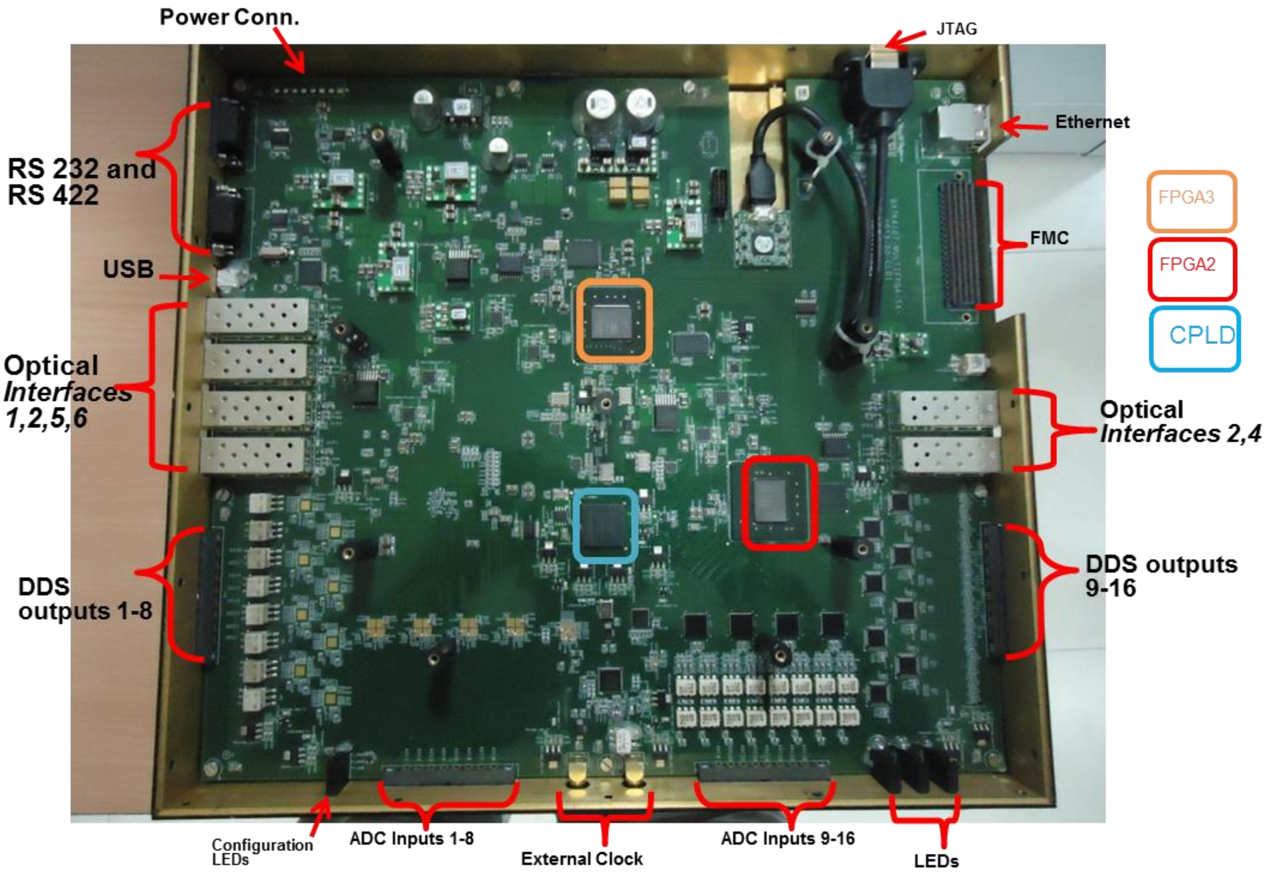


Figure 6-17: Hardware for Adaptive Beam Former VLSI Architecture

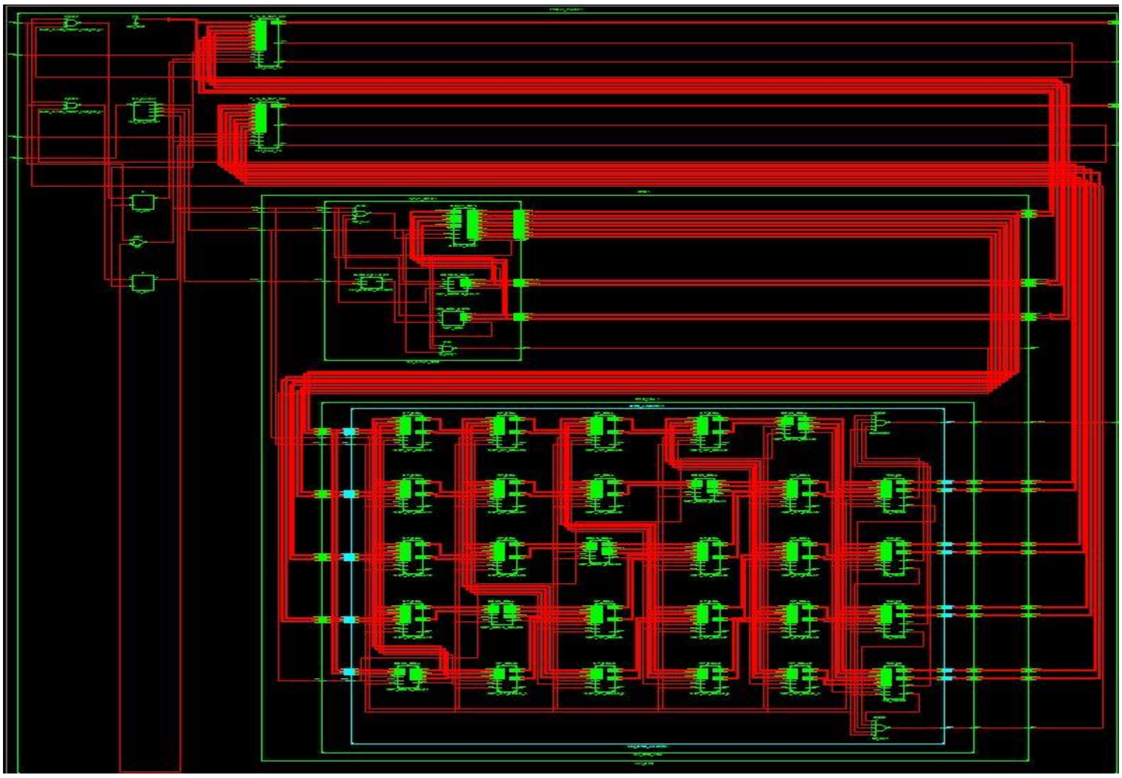


Figure 6-18: RTL view of four element IQRD RLS weight computation.

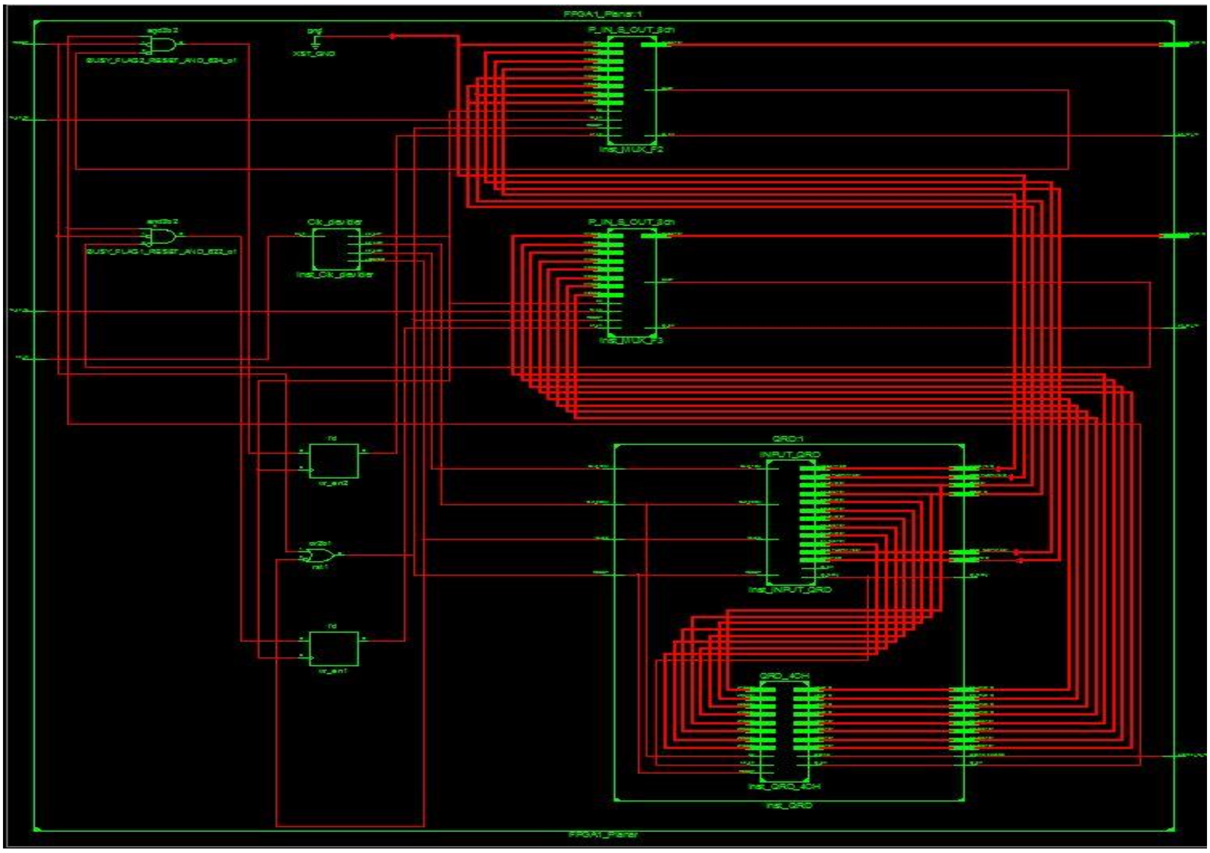


Figure 6-19: RTL view of eight element partial beam formation.

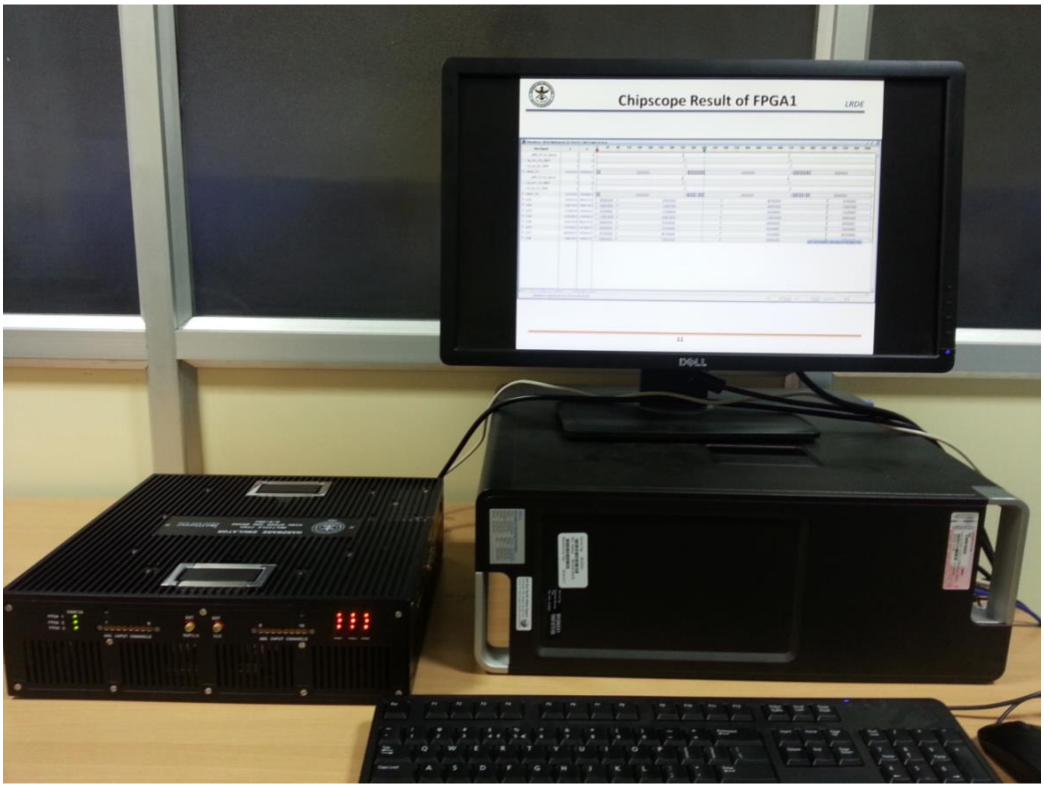


Figure 6-20: Hardware test set up for Adaptive Beam Formation.

6.4 Resource comparison

The FPGA based Adaptive Beam Former Architecture is designed using VHDL. The VLSI architecture modeling has been carried out and simulated. The functional verification has been done to validate the correctness of the results. Initially the 8 element linear array and 16 element linear array configuration is simulated. Then a planar array of 16 element configuration has been implemented. The implementation of ADBF for 1 beam is done on Kintex-7 series FPGA and later 2 beams also. Due to the requirement of more resources on hardware, for multiple beams, some reuses of the modules are necessary and hence time required calculating the multiple weights will be large. Table 6-1 and 6-2 describes the resource utilization summary.

Table 6-1: Resource utilization of FPGA-I and FPGA II for implementation of Inverse QRD RLS for Adaptive beam former of 8 elements with Floating point operations

Resource Utilization	Consumed	Available	Utilization in Percentage
Slice Registers total numbers	119840	508400	23%
Slice LUTs total numbers	144126	254200	56%
LUT-FF pairs	82594	181372	45%
Bonded IOBs total numbers	73	500	14%
Block RAM/FIFO total numbers	8	795	1%
BUFG/BUFGCTRLs total numbers	5	32	15%
DSP48E1s total numbers	643	1540	41%

Table 6-2: Resource utilization of FPGA-III for implementation of Adaptive beam former for sixteen elements. (Floating point)

Resource Utilization	consumed	Available	Utilization in percentage
Slice Registers total Numbers	16316	508400	3%
Slice LUTs total numbers	13396	254200	5%
LUT-FF pairs total numbers	10776	18936	56%
Bonded IOBs total numbers	66	500	13%
Block RAM/FIFO total numbers	1	795	0%
BUFG/BUFGCTRLs numbers	1	32	3%

Table 6-3: Resource utilization of FPGA-I and FPGA II for implementation of Inverse QRD RLS for Adaptive beam former of 8 elements with fixed point operations.

Logic Utilization	Used	Available	Utilization
Slice Registers Number	119840	508400	0%
Slice LUTs Number	144126	254200	0%
Fully used LUT-FF pairs	82594	181372	72%
Bonded IOBs Number	73	500	0%
Block RAM/FIFO Number	8	795	1%
BUFG/BUFGCTRLs Number	5	32	12%
DSP48E1s Number	643	1540	1%

Table 6-4: Resource utilization of FPGA-III for implementation of Adaptive beam former for sixteen elements with fixed point operations.

Logic Utilization	Used	Available	Utilization
Slice Registers Number	16316	508400	0%
Slice LUTs Number	13396	254200	0%
Fully used LUT-FF pairs Number	10776	18936	48%
Bonded IOBs Number	66	500	13%
Block RAM/FIFO Number	1	795	0%
BUFG/BUFGCTRLs Number	1	32	12%

Table 6-5: The Virtex-V FPGA resources utilization to form four beams.

Logic Utilization	Used	Available	Utilization
Slice Registers Number	15705	81920	19%
Slice LUTs Number	10525	81920	12%
DSP48E1s Number	306	320	95%
Block RAM/FIFO Number	69	298	24%

Table 6-6: The Virtex-VI FPGA resources utilization for four beam architecture

Logic Utilization	Used	Available	Utilization
Slice LUTs Number	86983	150720	57%
DSP48E1s Number	634	768	82%
Block RAM/FIFO Number	50	416	12%
BUFG/BUFGCTRLs Number	6	32	18%
PLL_ADVs	1	12	8%

Table 6-7: Resource utilization Comparison for 32 bit Fixed and Floating point operations.

S.No	Parameter	32 –bit Fixed Point	32 bit-Floating Point
1	Slice Registers	40%	23%
2	LUTs	68%	45%
3	DSP	26%	41%
4	Memory	25%	19%

6.5 Summary

In this chapter hardware realizations using the multiple FPGAs were covered. The multiple beam former architecture is developed initially with Virtex-V FPGA considering the four element array and the weights were calculated offline and stored in the memory of this board. The architecture is extended further to a sixteen element planar array and the optimal weights were calculated online and multiple beams were formed using the Virtex-VI FPGA configuration board. The optimized architecture was not meeting the processing time the architecture was split into three FPGAs and the architecture is redesigned to form multiple adaptive digital beams using fixed point arithmetic and floating point arithmetic operations. The resource utilization for all the architectures is compared and is tabulated.

In the next chapter all the experimental results are covered in detail. The adaptive beam former architecture designed and simulated in the hardware simulators and these results are explained. The architecture further realized with the target device and actual results are captured and explained. The real time data is taken and plotted in a simulator and found that all the results are as expected.

Chapter 7

Experimental Results and Discussions

Chapter 7.

Experimental Results and Discussions

7.1 Introduction

There are three different types of digital beam former architecture have been developed. Each one of the architecture is designed, simulated, verified and validated on the FPGA based proto hardware. Each one of the architecture results are presented in this section. The architecture developed is modular in nature and the module level simulation has been completed and further the similar results are obtained on the hardware.

Adaptive weight computation module is simulated for 16 element array using the QRD RLS algorithm and the same results are as shown in Figure 7.1.

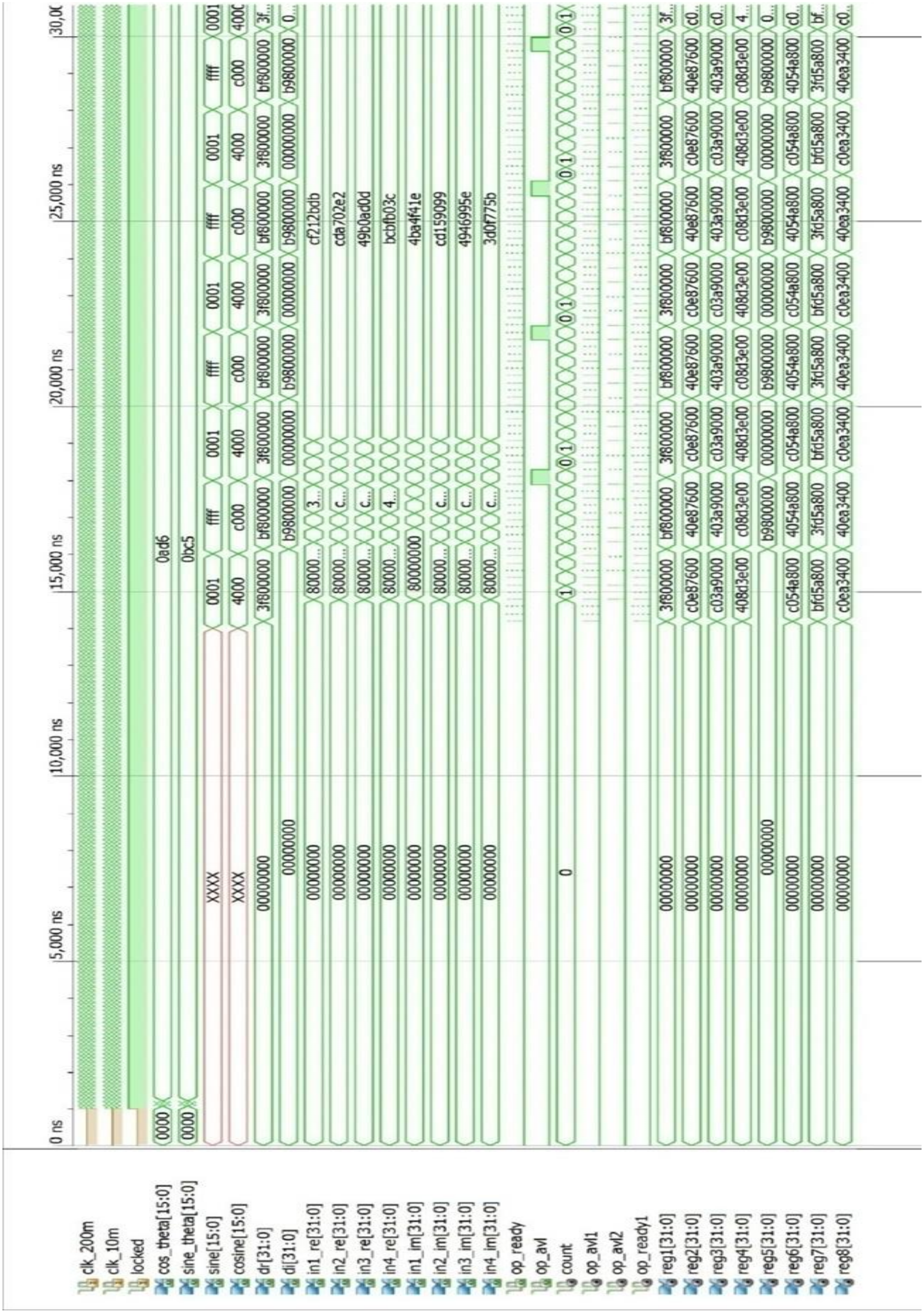


Figure 7-1: Calculation of Adaptive Weights for 16 Element Array at timing simulation.

The efficient tools are used to do the simulation and architecture is optimized using many techniques like re-timing so that the computation time and delays can be optimized in the parallel and pipelined processing approach.

The Adaptive weight calculation module for 16 element array is designed and timing has been verified on the chip as shown in the Figure 7-2.

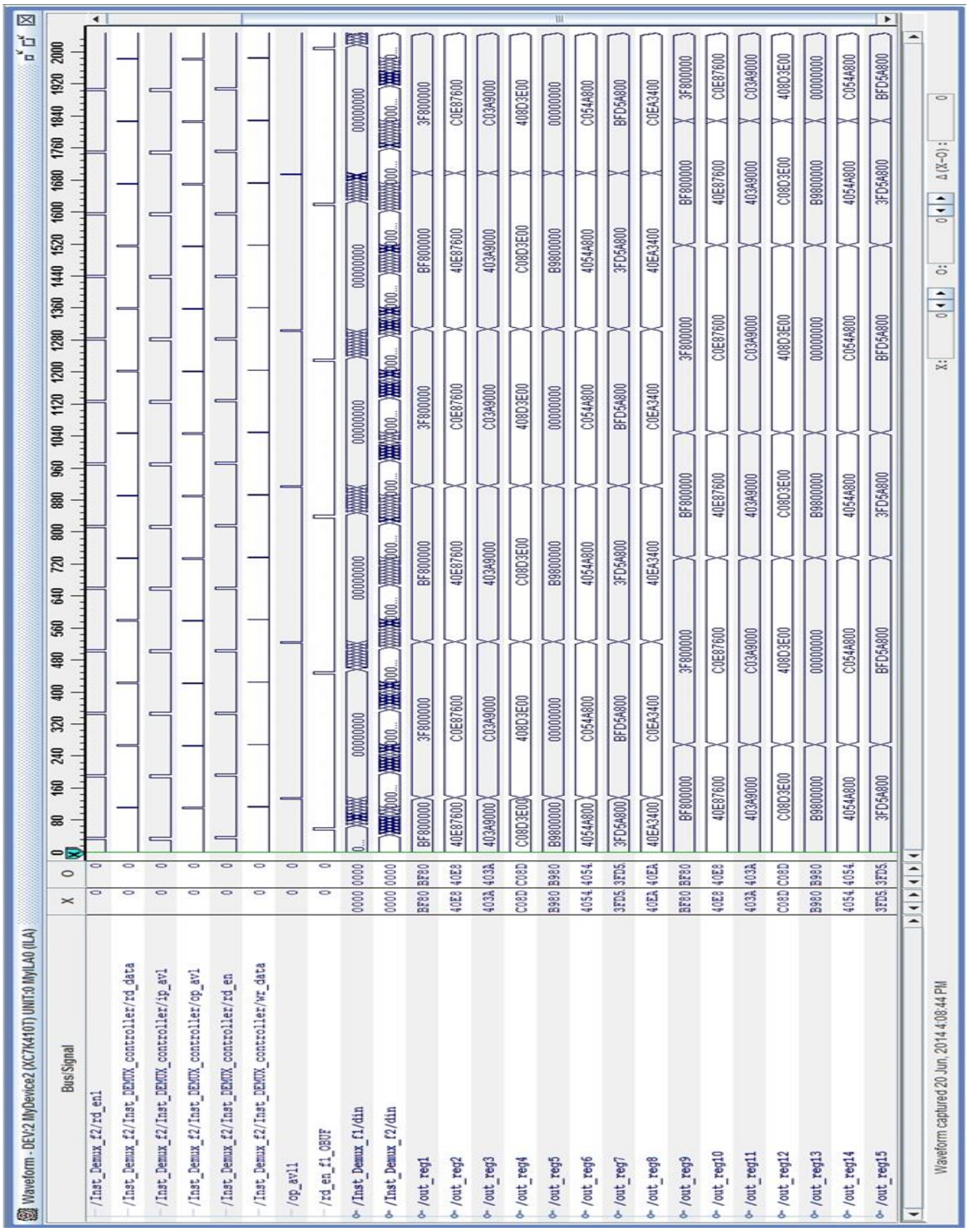


Figure 7-2: Adaptive Weight calculation for 16 Element Array at Chip level.

The architecture developed was simulated with 32 bit fixed point arithmetic as shown in Figure 7-3. The weights were calculated for various look angles for 16 element phased array.

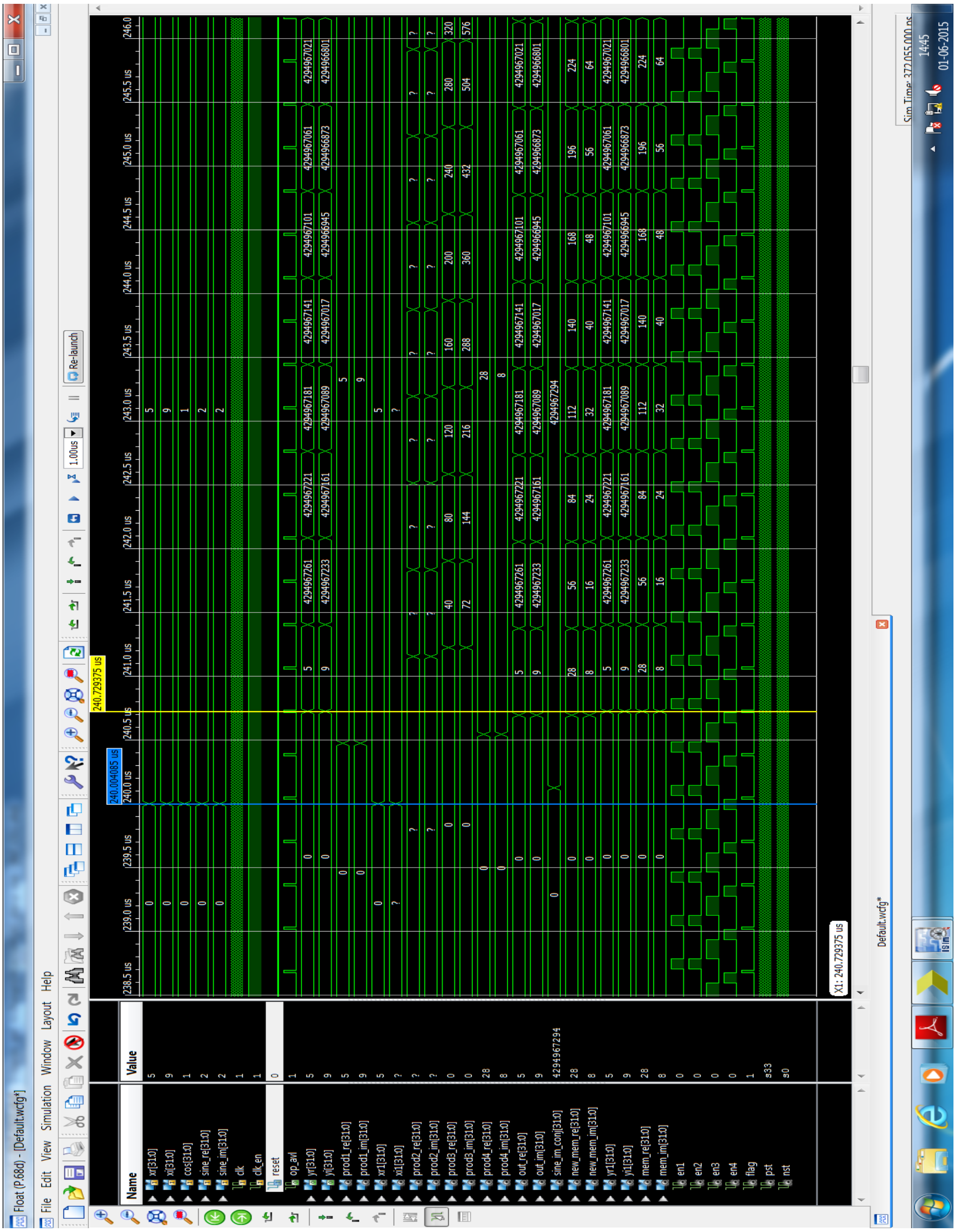
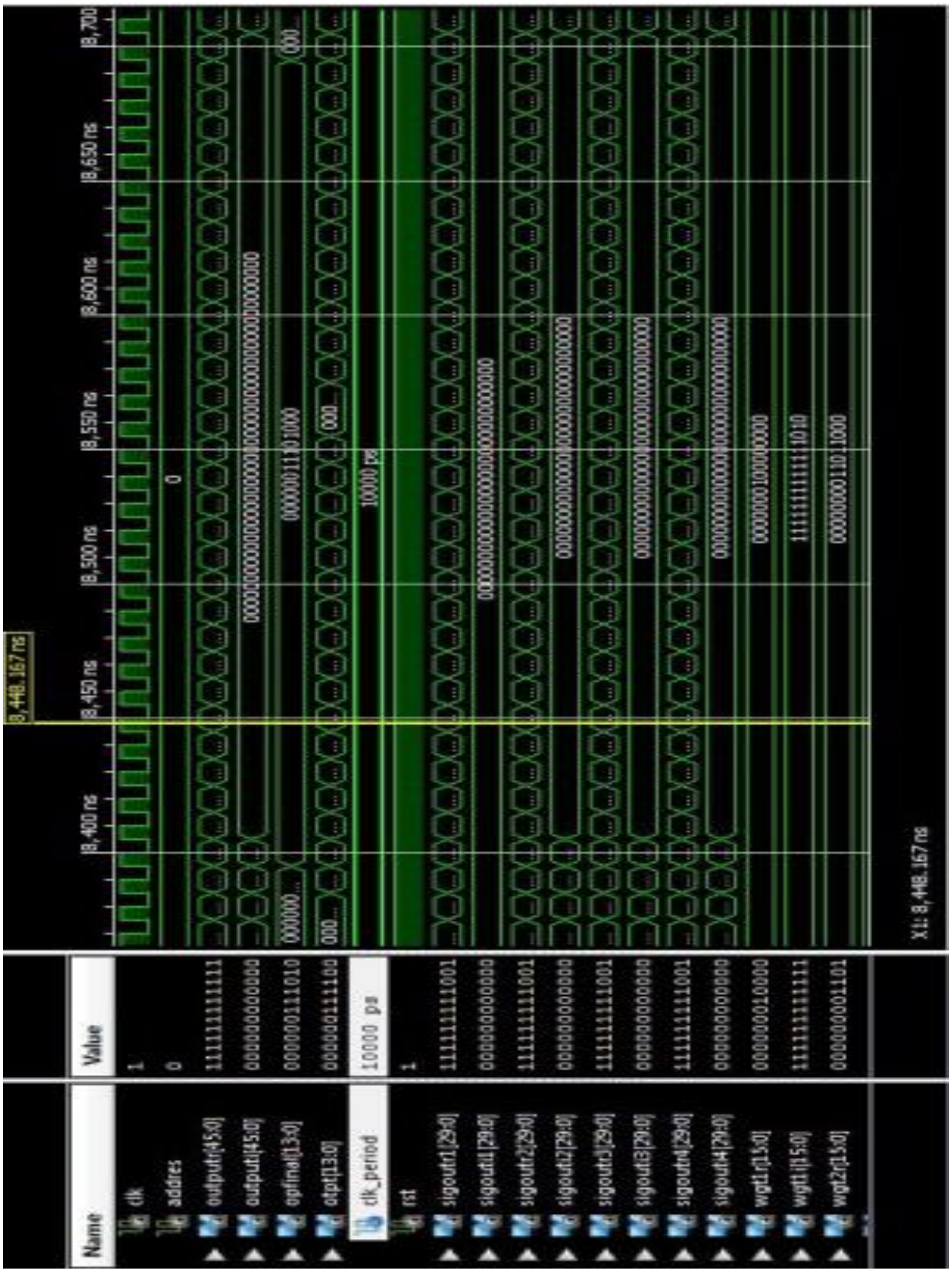


Figure 7-3: Computation of Adaptive Weights for 16 Element Array functional simulations.

- Fixed point 32 bit input values the total time for the first out is 720 ns
- All the output values are rounded off to 32 bit instead of 64 bit.
- This is carried out in order to prevent the bit growth.
- This has resulted into great reduction of resource utilization of FPGA.

- The optimal weights are available after 720 ns instead of 800 ns

Results of fixed weight beam former architecture are shown in the Figure 7-4. The weights are stored in the RAM and the down converted IF signal is input from the ADC is fed to the Architecture. It has generated multiple beams.



Name	Value
clk	1
addres	0
outputr[45:0]	111111111111
outputi[45:0]	000000000000
opfinal[13:0]	000000111010
otpt[13:0]	000000111100
clk_period	10000 ps
rst	1
sigoutr1[29:0]	111111111001
sigouti1[29:0]	000000000000
sigoutr2[29:0]	111111111001
sigouti2[29:0]	000000000000
sigoutr3[29:0]	111111111001
sigouti3[29:0]	000000000000
sigoutr4[29:0]	111111111001
sigouti4[29:0]	000000000000
wgt1r[15:0]	000000010000
wgt1i[15:0]	111111111111
wgt2r[15:0]	000000001101

Figure 7-4: Simulation of digital down converter and multiple beams

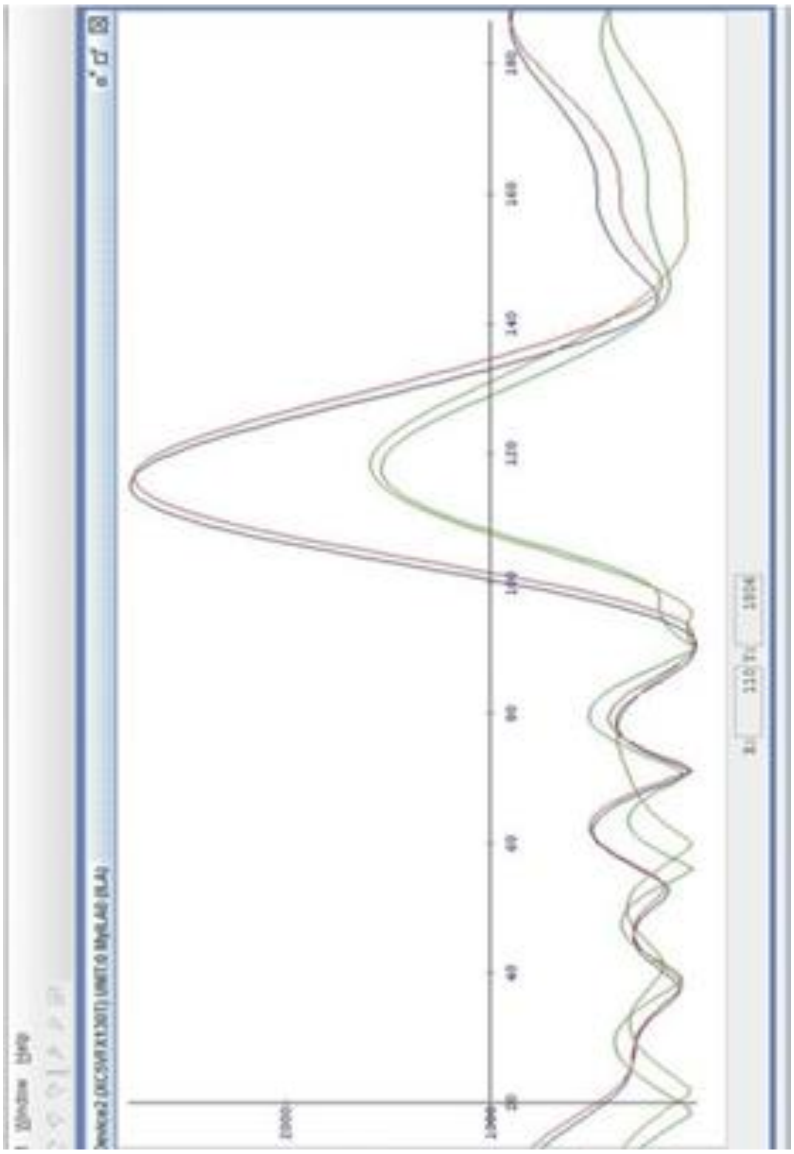


Figure 7-5: Real time data captured from the FPGA and plotted in chipscope tool.

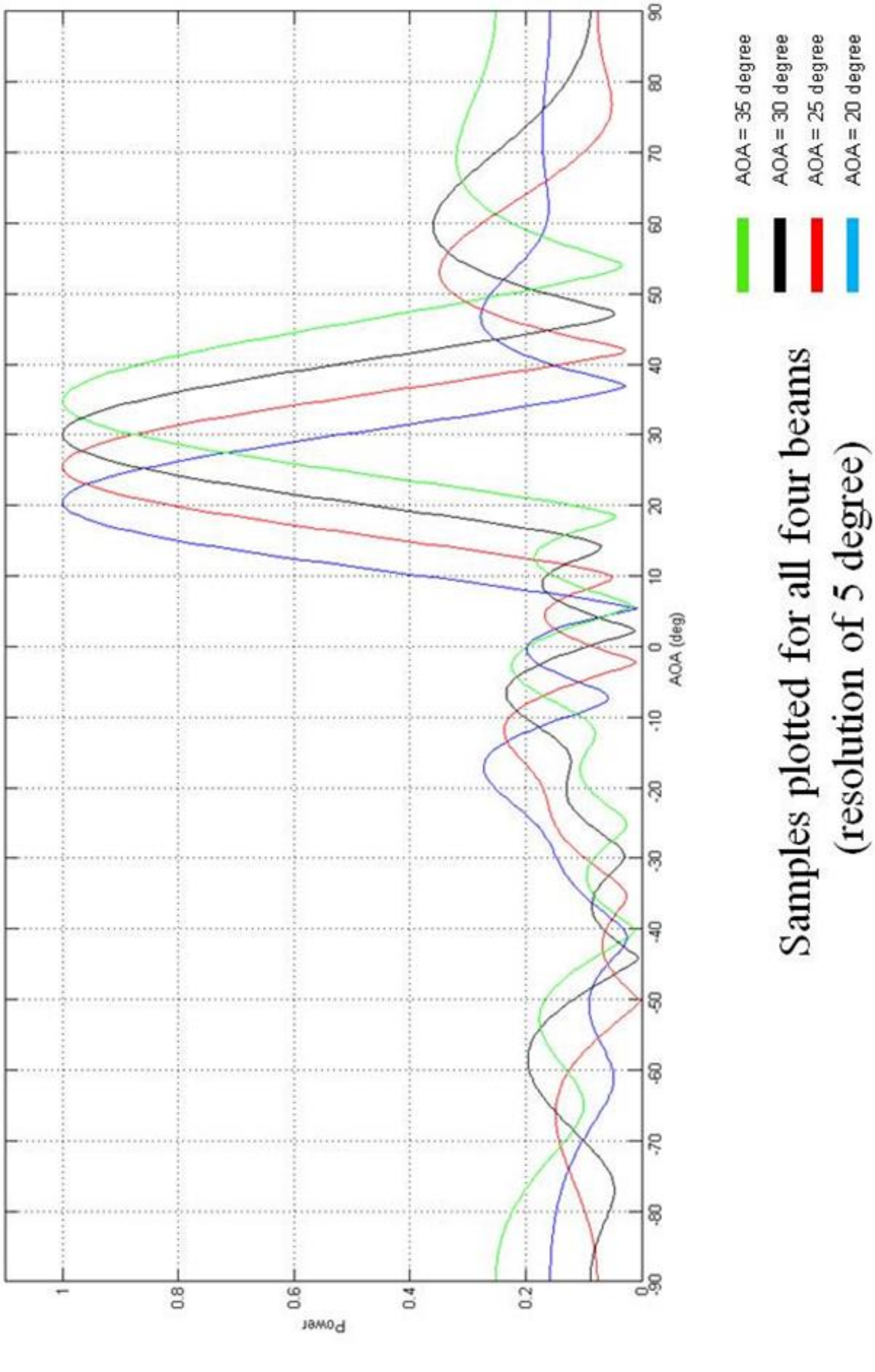


Figure 7-6: The real time data is received from the prototype board and MATLAB is used to plot the multiple beams.

7.2 Results of Adaptive and fixed weights beam former architecture

1) The fixed weight beam forming architecture is developed for a planar array size of sixteen elements antenna array (four by four matrix)

- Direction of Arrival in Azimuth=0^0 and in Elevation=20^0
- Number of beams formed is Four
- Azimuth and Elevation angle is variable
- Angle by which beams are separated=100
- The beams are formed for azimuth angles= - 20 deg , -10 deg, 0 deg, 10 deg

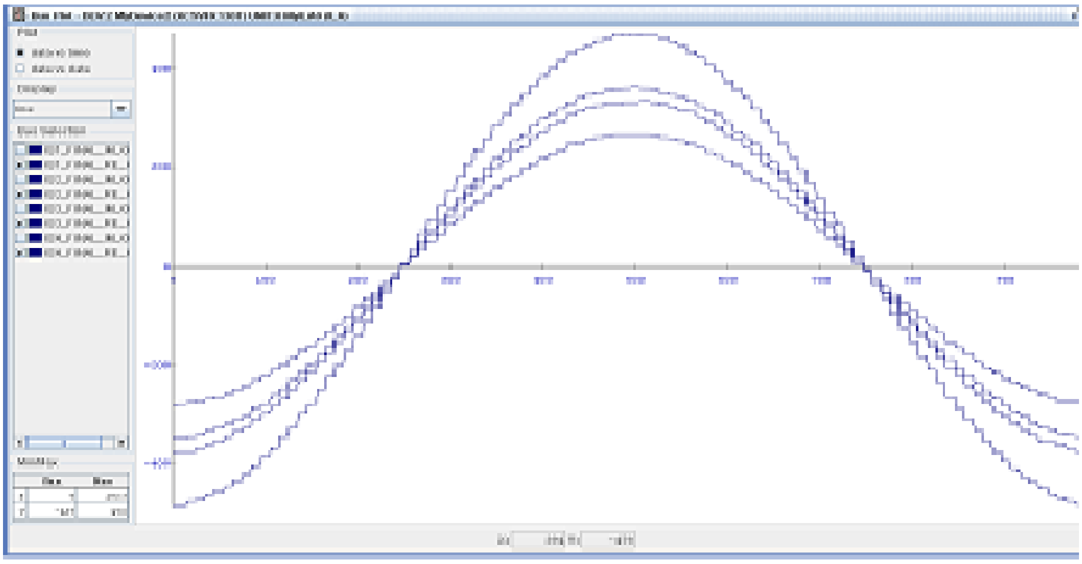


Figure 7-7: The planar antenna array Inphase signal output indicating four digital beams in the chipscope.

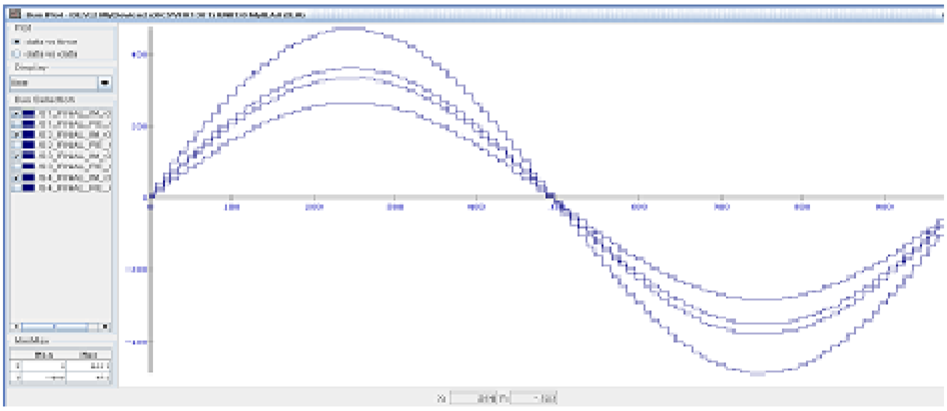


Figure 7-8: The planar antenna array quadrature phase signal output indicating four digital beams in the chipscope.

The Weights are imported from FPGA based proto hardware to MATLAB and multiplied with scanning vector to form multiple beams in the desired direction.

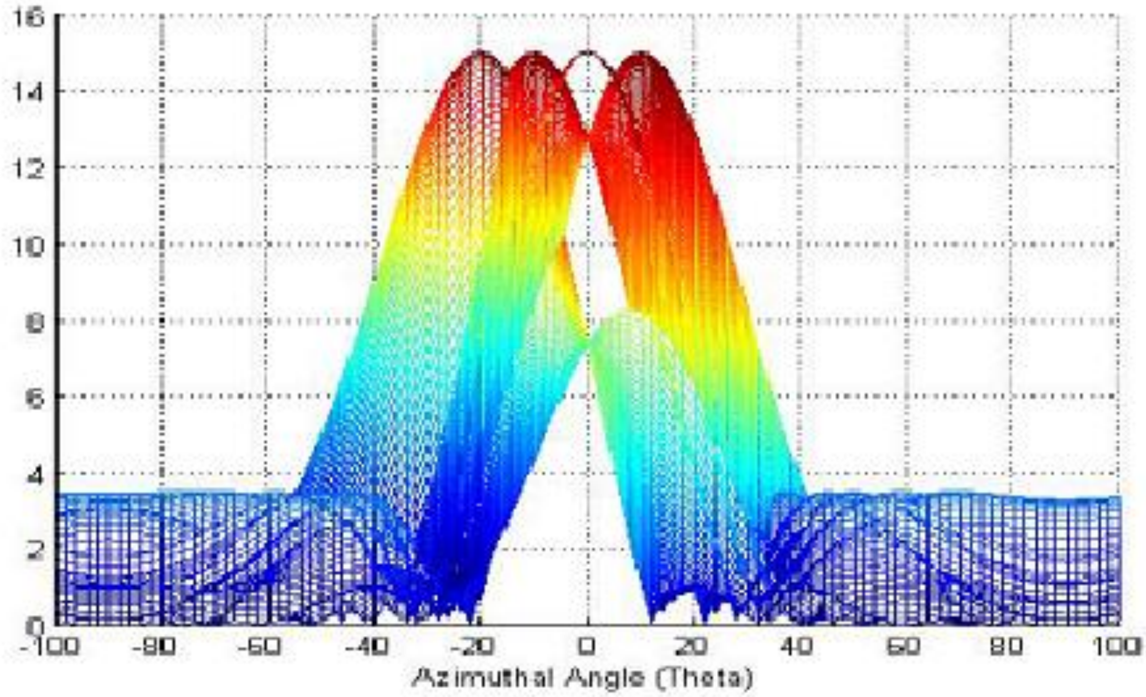


Figure 7-9: Four digital beams plotted in MATLAB using the weight vectors stored inside FPGA in the desired direction at Azimuth zero deg and Elevation 20 deg.

2) Following are the Chipscope outputs for fixed beamforming for 16 element planar antenna array.

- Direction of arrival Azimuth=40 deg and Elevation=10 deg
- Number of beams formed is four
- Elevation angle is variable and Angle by which beams are separated is 10 deg.
- The beams are formed for elevation angles= -10 deg, 0 deg, 10 deg, 20 deg

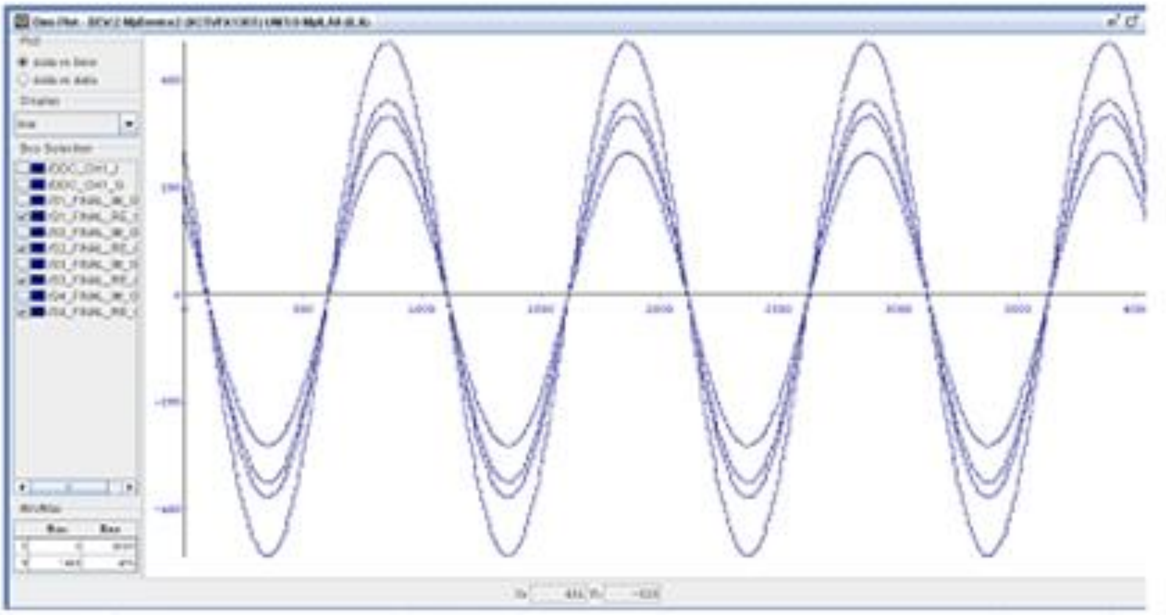


Figure 7-10: Planar Antenna array output in the receive mode showing four in phase signals corresponding to four beams.

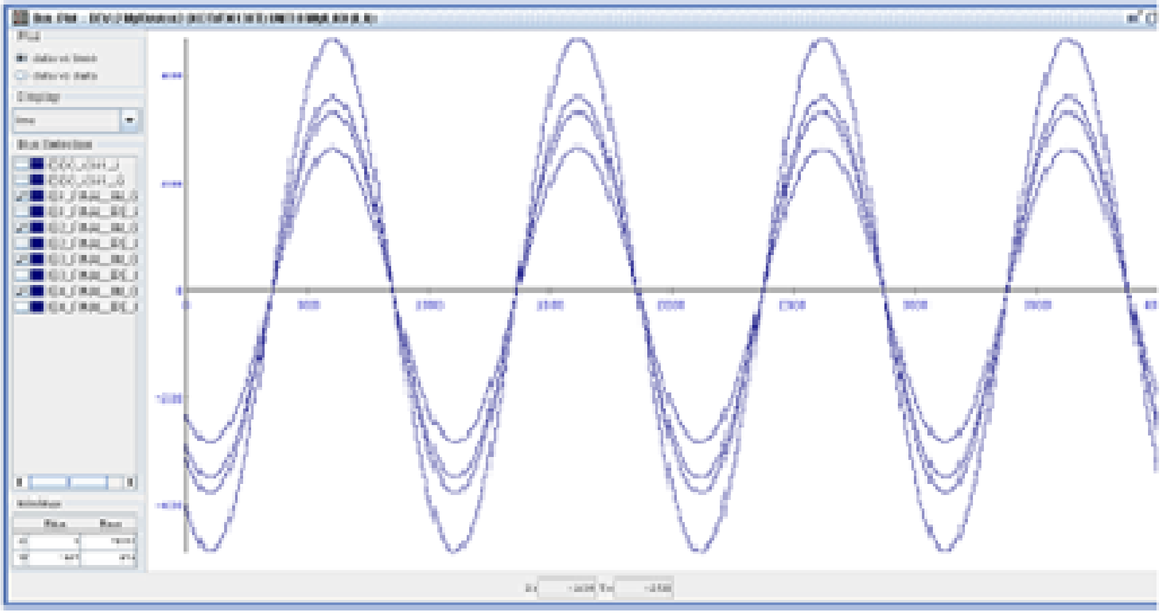


Figure 7-11: Planar Antenna array output in the receive mode showing four quadrature phase signals corresponding to four beams.

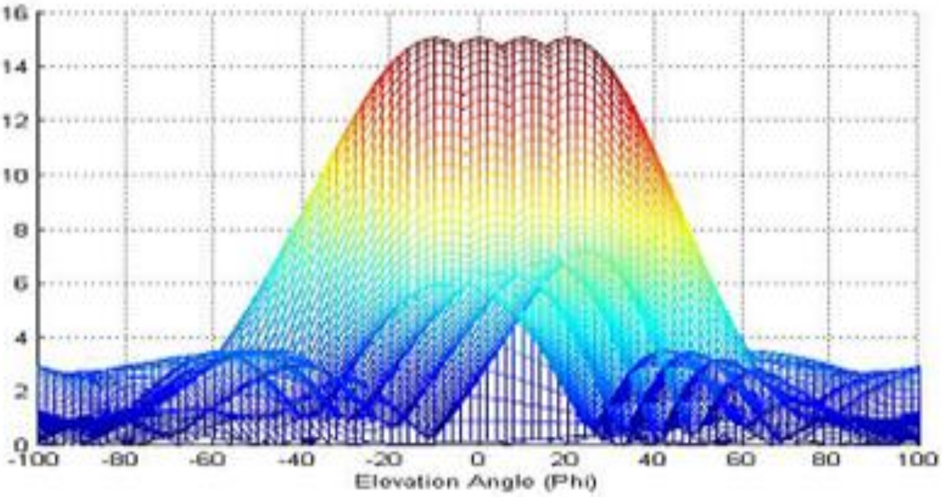


Figure 7-12: Four beams plotted in MATLAB using the weight vectors stored inside FPGA in the direction of arrival at Azimuth 0 deg and Elevation 20 deg.

3) Following are the Chipscope outputs for fixed beamforming for 16 element planar antenna array consisting of four by four matrix.

- Direction of arrival is in Azimuth = 0 deg and Elevation=30 deg
- Number of multiple beams formed is nine.
- Azimuth angle is variable and Angle by which each beams are separated is 10 deg.
- The beams are formed for azimuth angles viz, -40 deg ,-30 deg, -20 deg , -10 deg, 0 deg , 10 deg , 20 deg ,30 deg , 40 deg.

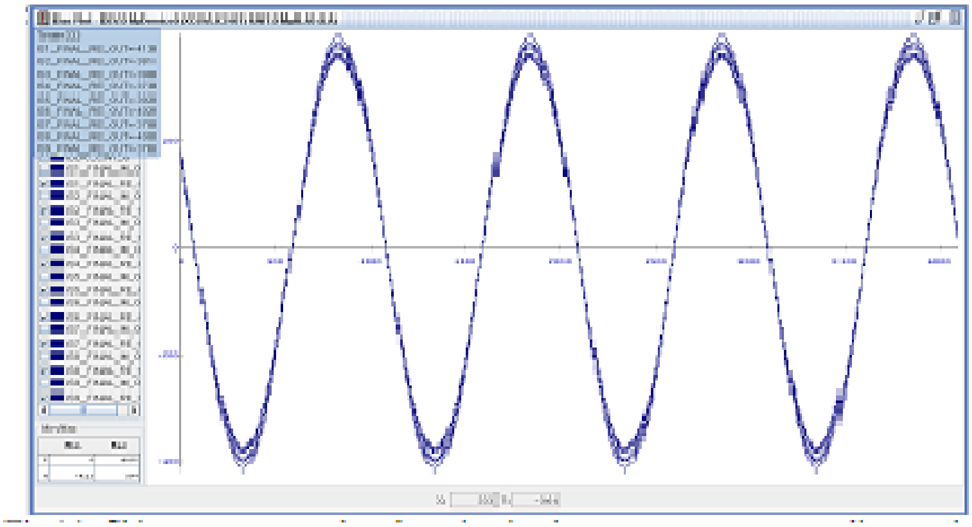


Figure 7-13: Chipscope output showing nine in phase outputs corresponding to nine beams for 16 elements planar antenna array.

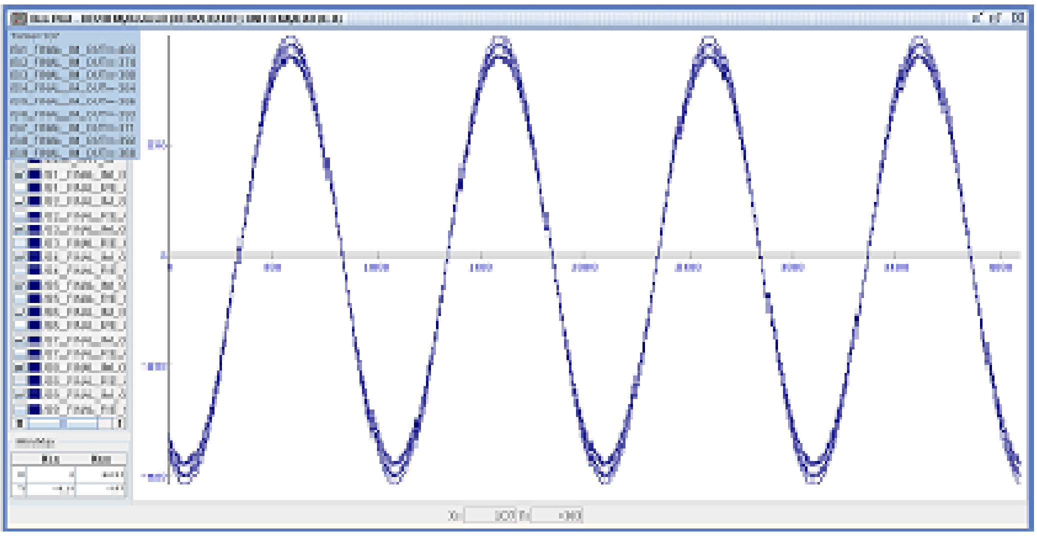


Figure 7-14: Chipscope output showing nine quadrature phase outputs corresponding to nine beams for 16 element planar antenna array

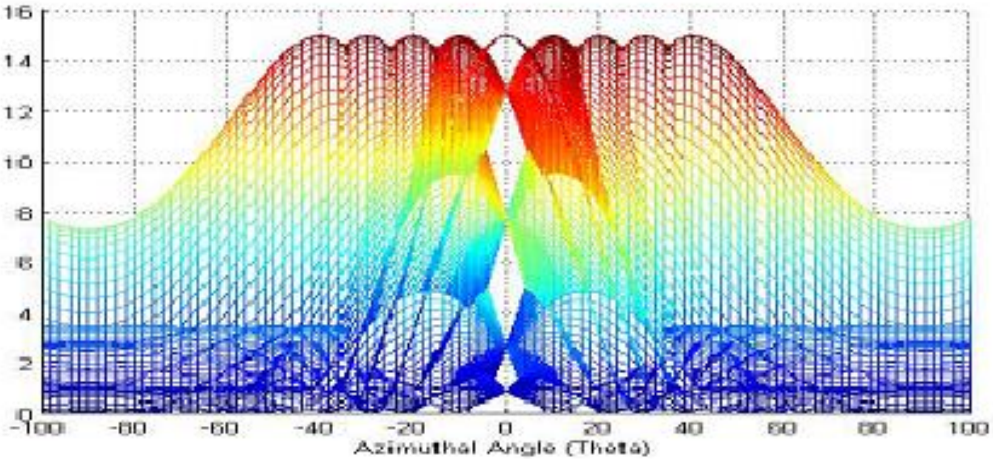


Figure 7-15: Nine beams plotted in MATLAB using the weight vectors stored inside FPGA.

4) Following are the Chipscope outputs for fixed beamforming for 16 element planar antenna array.

- Direction of Arrival in Azimuth=30 deg and Elevation=0 deg.
- Number of multiple beams formed is 09
- Elevation angle is variable and Angle by which beams are separated is 10 deg
- The beams are formed for elevation angles = -40 deg, -30 deg, -20 deg, -10 deg, 0 deg, 10 deg, 20 deg, 30 deg, 40 deg.

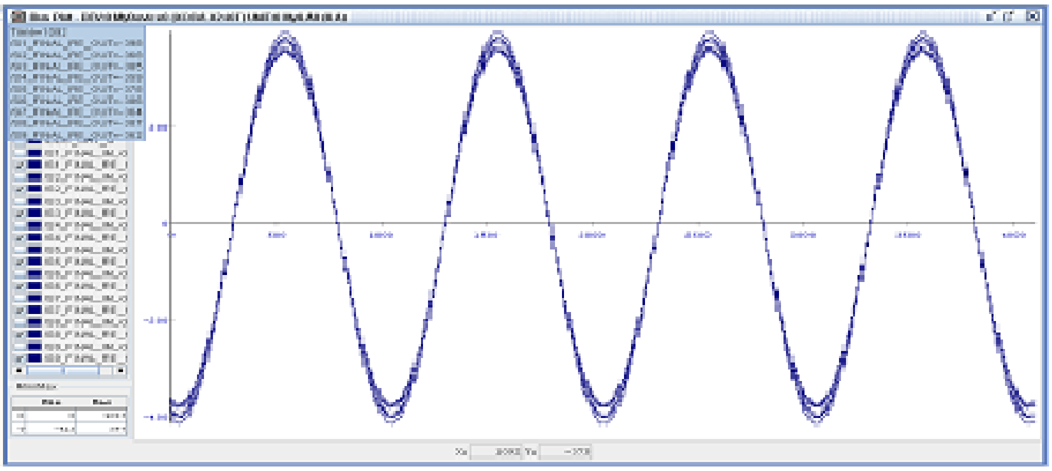


Figure 7-16: Chipscope output showing nine in phase output corresponding to nine beams for 16 element planar antenna array.

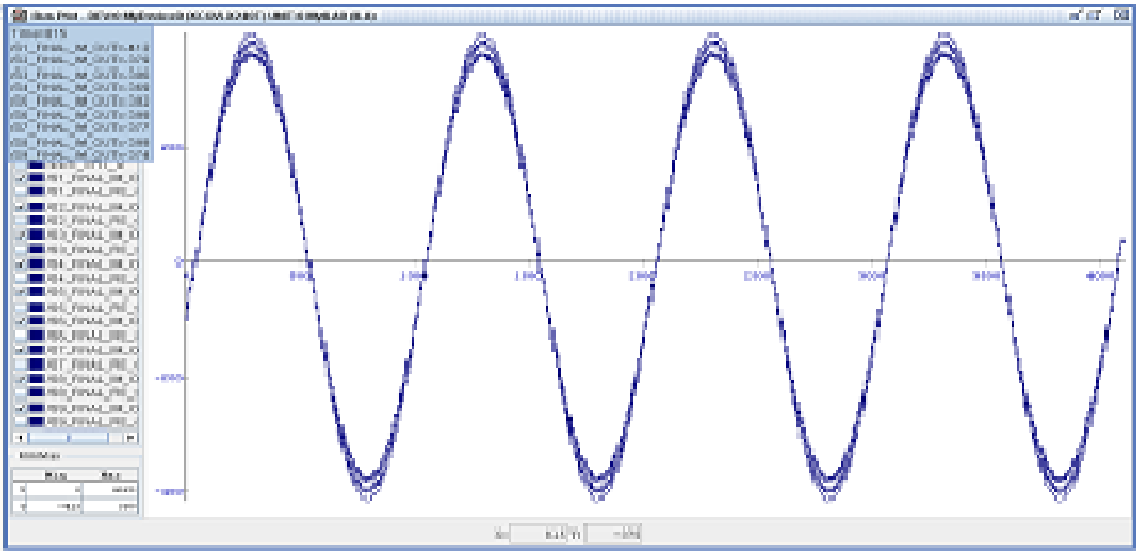


Figure 7-17: Chipscope output showing nine quadrature phase outputs corresponding to nine beams for 16 element planar antenna array

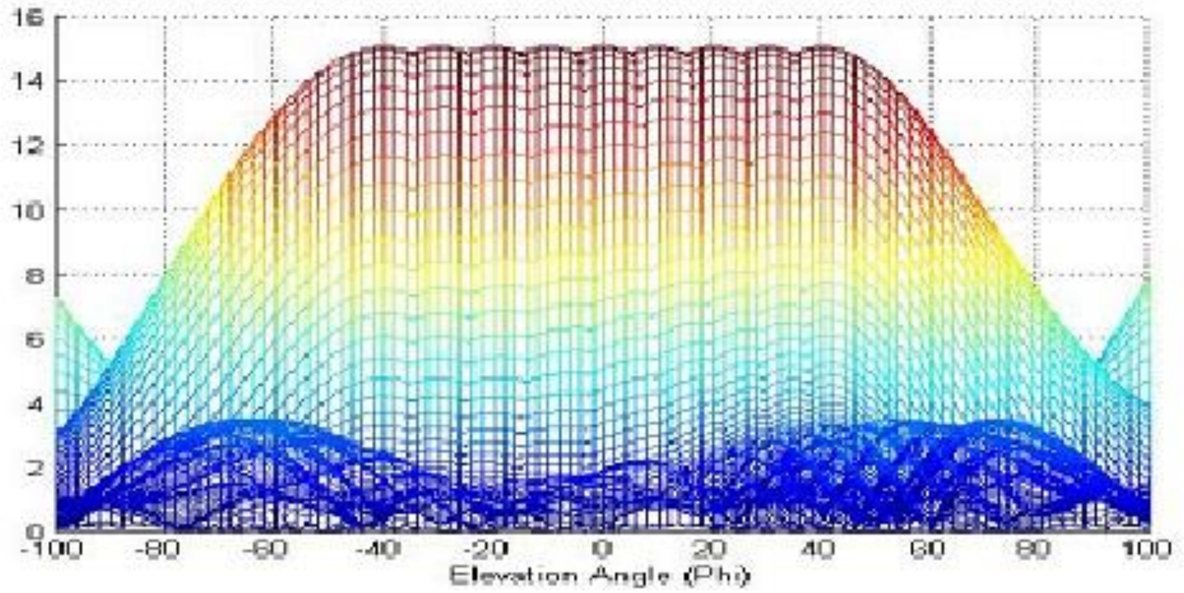


Figure 7-18: Nine beams plotted in MATLAB using the weight vectors stored inside FPGA.

Following is the result of the implementation of adaptive beam forming using inverse QRD-RLS algorithm for 16 element planar phased array radar in MATLAB.

- Input azimuth angle of arrival for desired signal = -30 Deg
- Input elevation angle of arrival for desired signal = 40 Deg

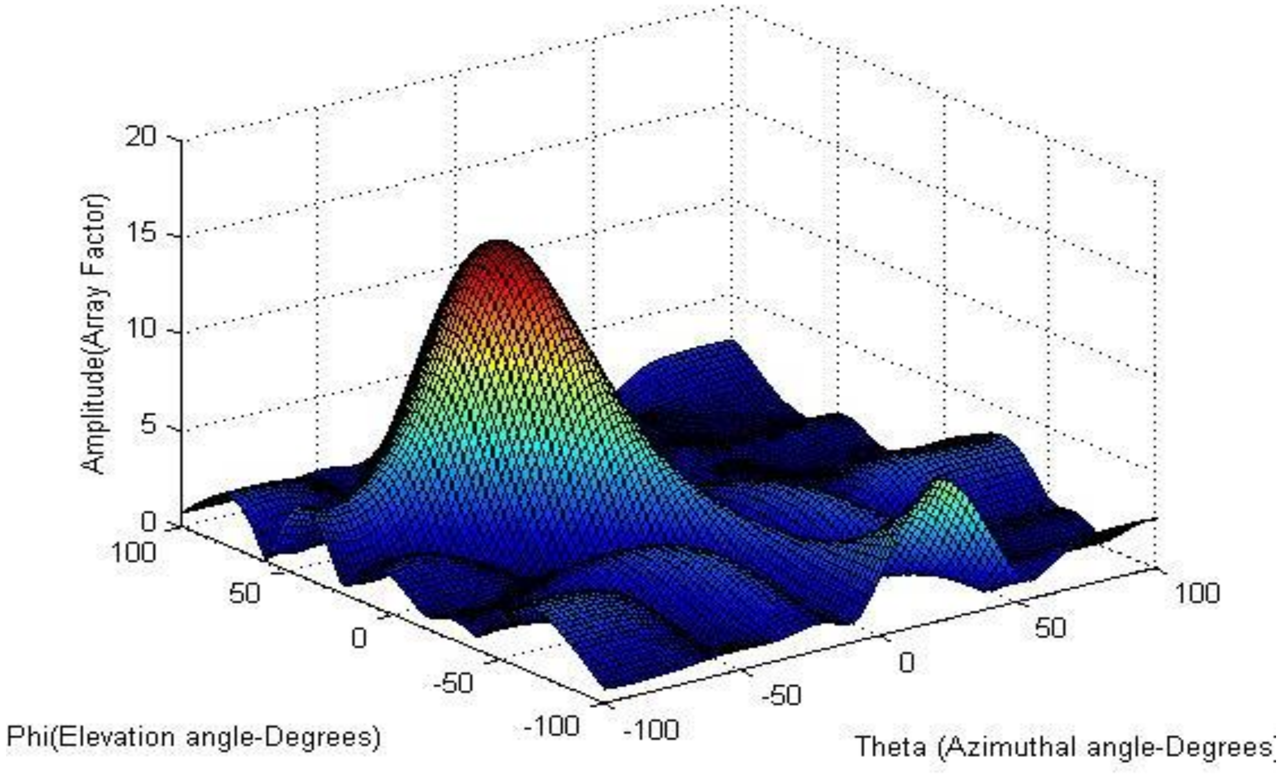


Figure 7-19: One Adaptive digital Beam plot for sixteen element planar antenna array using weights generated by inverse QRD-RLS in MATLAB.

Following is the result of the implementation of adaptive beam forming using conventional QRD-RLS algorithm for 16 element planar phased array radar in MATLAB.

- Input azimuth angle of arrival for desired signal =40 Deg
- Input elevation angle of arrival for desired signal =20 Deg

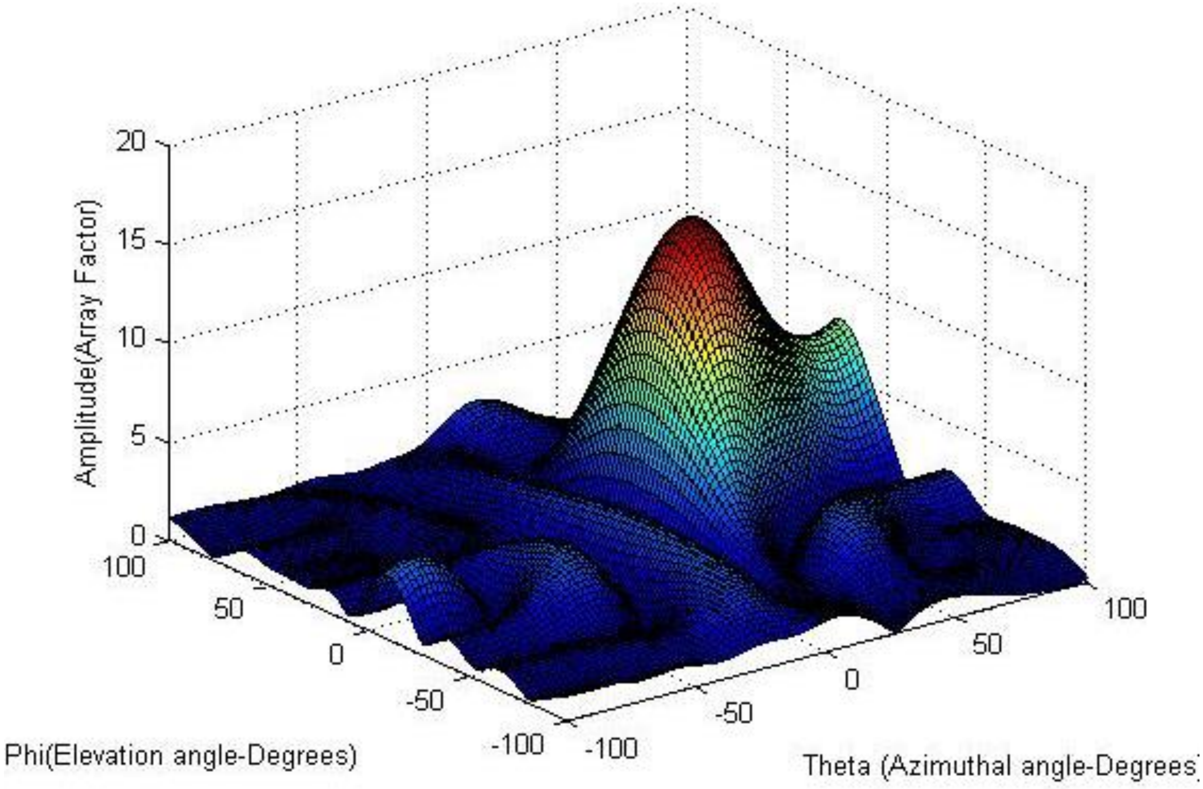


Figure 7-20: Beam plot for the 16 element planar antenna array using weights generated by conventional QRD-RLS in MATLAB.

7.3 Results of Adaptive Beam Former using multiple FPGA configuration

The entire VLSI architecture modeling has been carried out and the same has been simulated. The functional verification has been done to validate the correctness of the results. Initially the sixteen element linear array configuration is simulated, once the results were found satisfactory, and then a planar array of sixteen element configuration has been implemented. The implementation of ADBF for two beams is completed on Kintex-7 series FPGA due to the requirement of more resources for parallel processing. The prototype hardware consists of three FPGAs on board and input is given from DDS module using the IP core inside the FPGA which generates 60 MHz output of the ADC. Following is the result of the implementation online of adaptive weight computation using inverse QRD-RLS algorithm for Eight element planar array. The direction of arrival is considered azimuth angle of arrival for desired signal= -30 deg and input elevation angle of arrival for desired signal =10 deg.

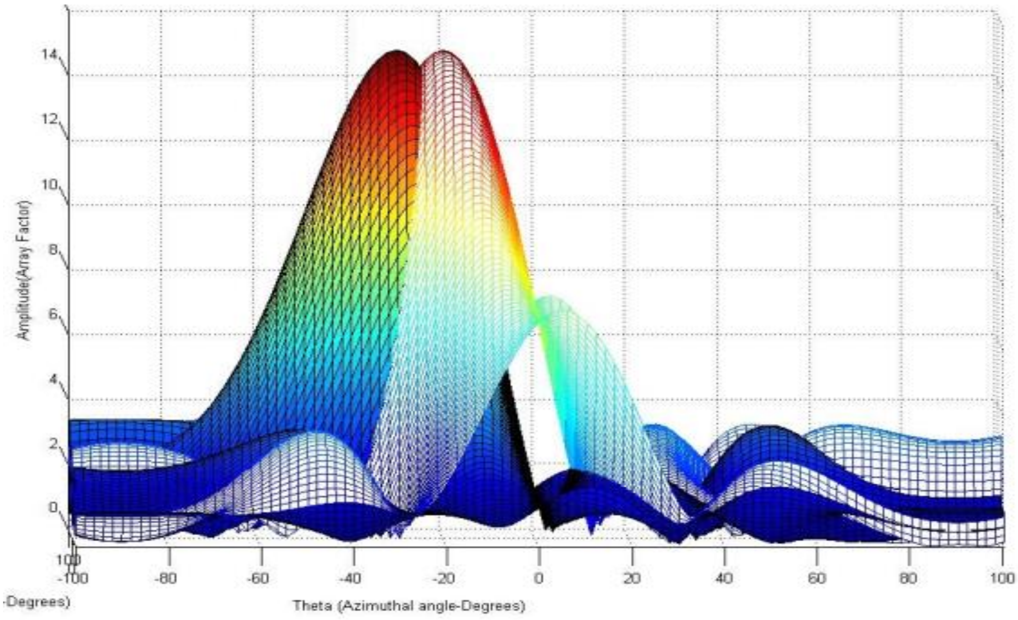


Figure 7-21: Eight elements planar array to form two adaptive beams from the hardware in the direction of arrival.

The developed architecture is extended to increase the number of radiating elements from eight elements to sixteen elements phased array configuration. Inverse QRD RLS algorithm is used compute the weights online and the data captured and plotted in the MATLAB. Input azimuth angle of arrival for desired signal is at 0 deg, Input elevation angle of arrival for desired signal is at 10 deg.

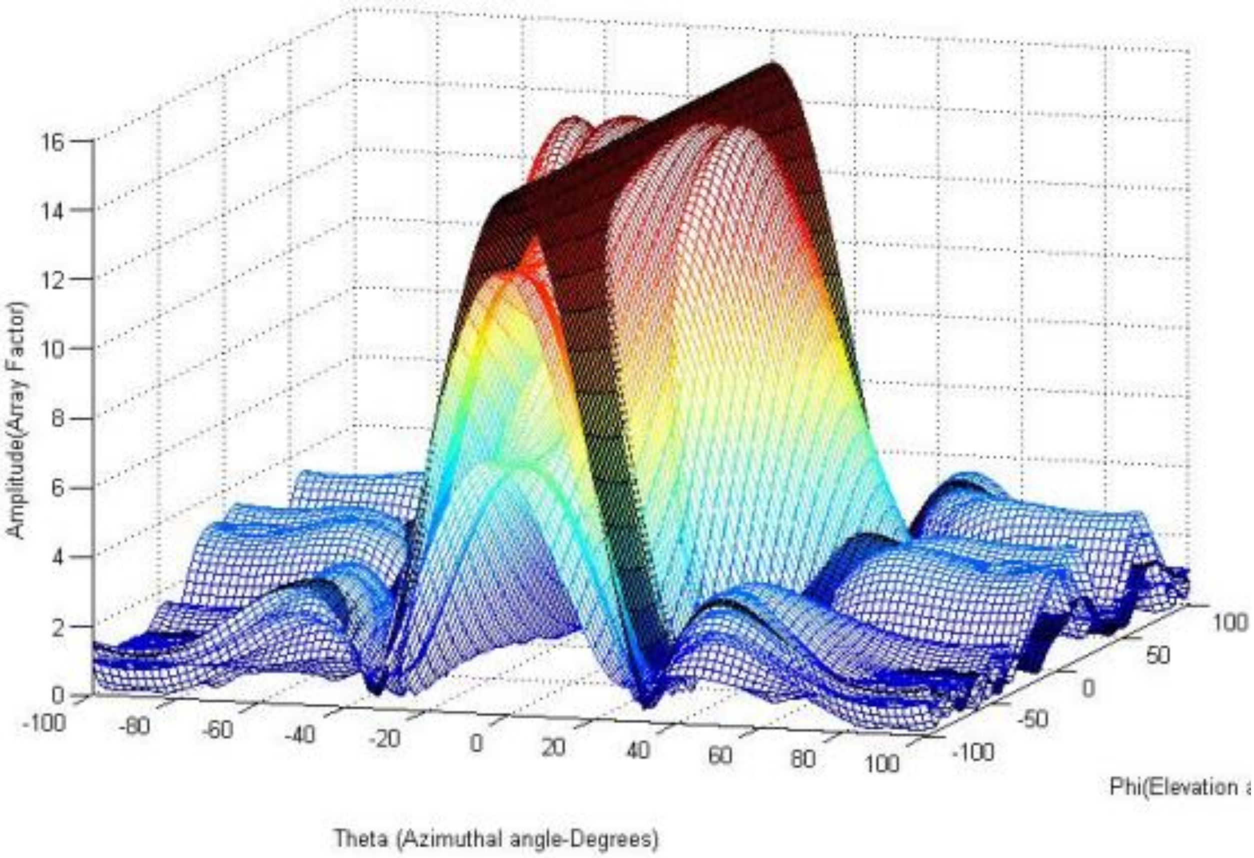


Figure 7-22: Multiple beams generated for sixteen elements planar array.

The architecture extended to form Nine Adaptive beams weights computation and beam formation. Sixteen elements array is considered and all the sixteen element data is received simultaneously and processed for weight computation and Adaptive beam formation. Azimuth angle of arrival for desired signal is 10 deg and elevation angle of arrival for desired signal is 25 deg.

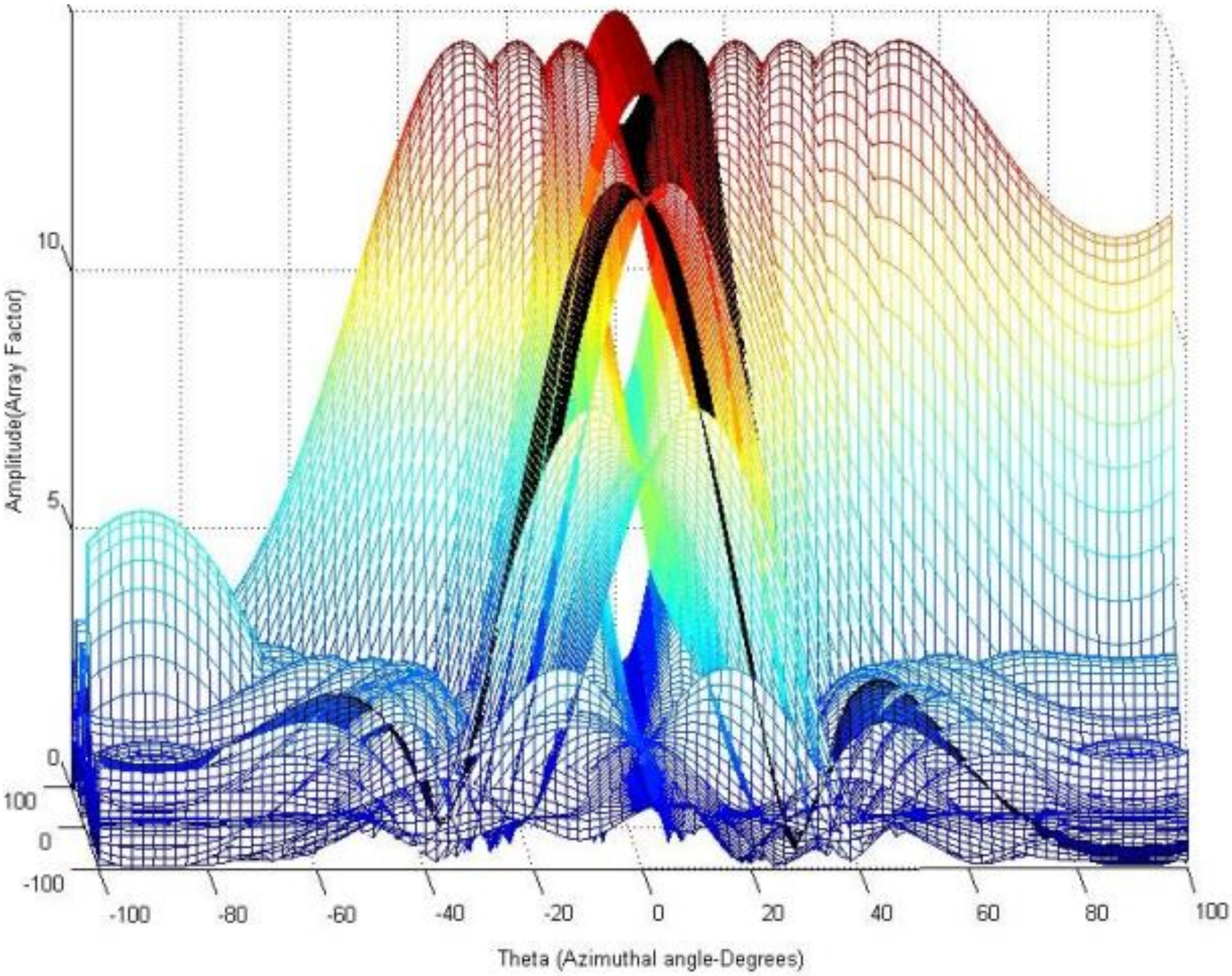


Figure 7-23: Planar array of sixteen elements Nine adaptive beams.

Following figure shows the result of the implementation of adaptive beam forming using Inverse QRD-RLS algorithm for 16 element linear array. Input angle of arrival for desired signal is 10 deg, and Interference angle is 40 deg.

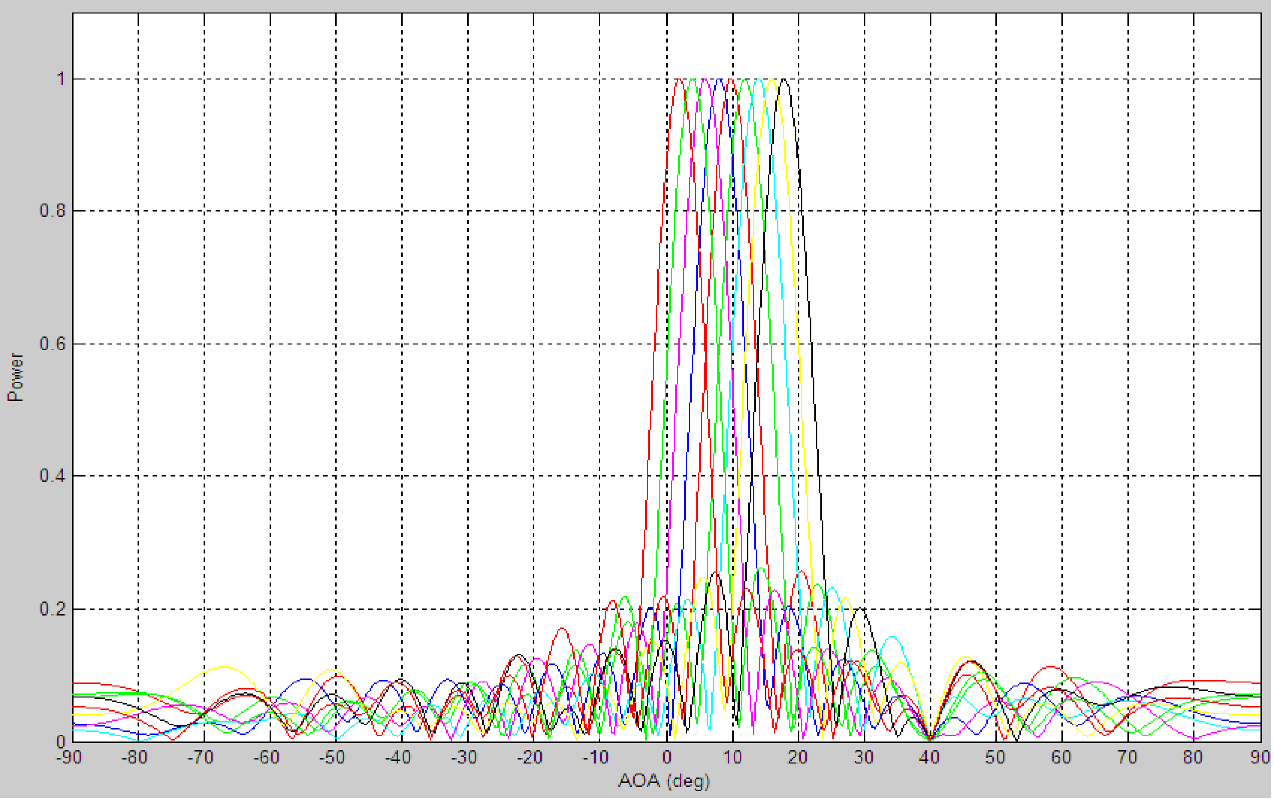


Figure 7-24: Linear array of sixteen elements Nine adaptive beams.

7.4 Results of Adaptive beams formed with Fixed point Operations

The Result of implementation of adaptive beam forming using Inverse QRD-RLS algorithm for 16 element planar array is shown in Figure 7-25. Angle of arrival for desired signal is 0 deg, and Interference angle is 30 deg, and single beam is generated.

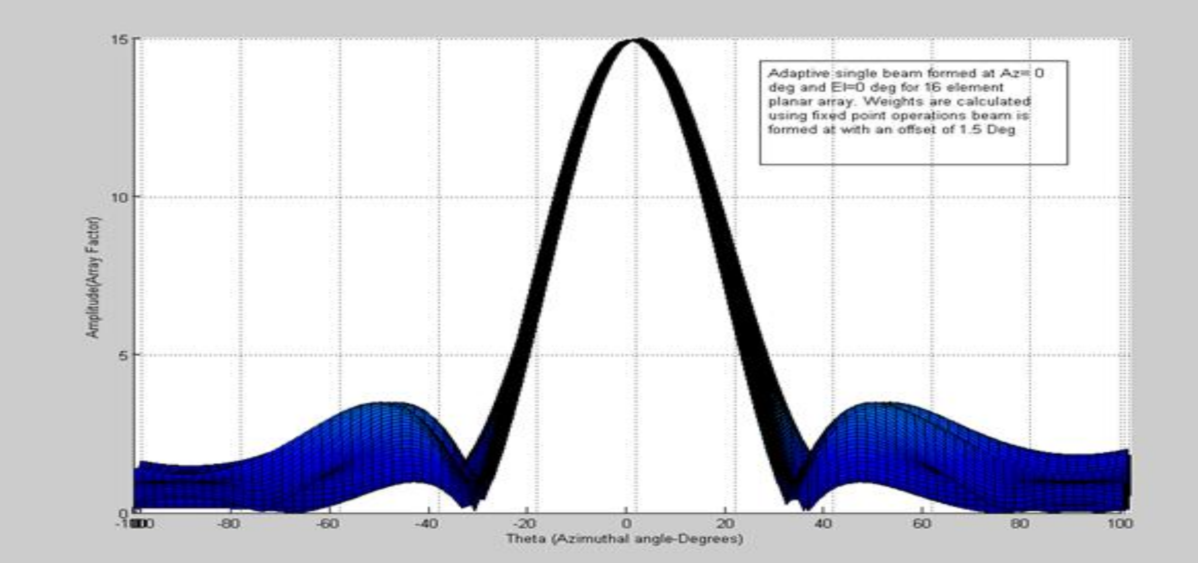


Figure 7-25: Planar array of sixteen elements single adaptive beam.

The Result of implementation of adaptive beam forming using Inverse QRD-RLS algorithm for 16 element planar array is shown in Figure 7-26. Angle of arrival for desired signal = 0 deg and Interference angle =30 deg, two beams are produced.

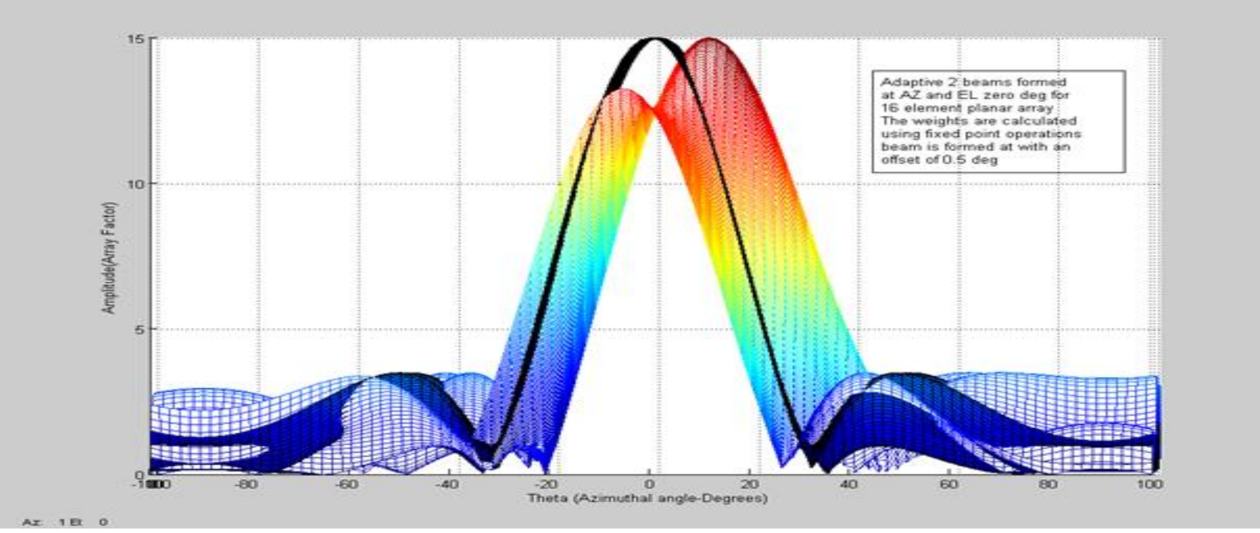


Figure 7-26: Planar array of sixteen elements two adaptive beam.

The Result of implementation of adaptive beam forming using Inverse QRD-RLS algorithm for 16 element planar array is shown in Figure 7.27. Angle of arrival for desired signal = 0 deg, Interference angle = 30 deg, Five beams are produced.

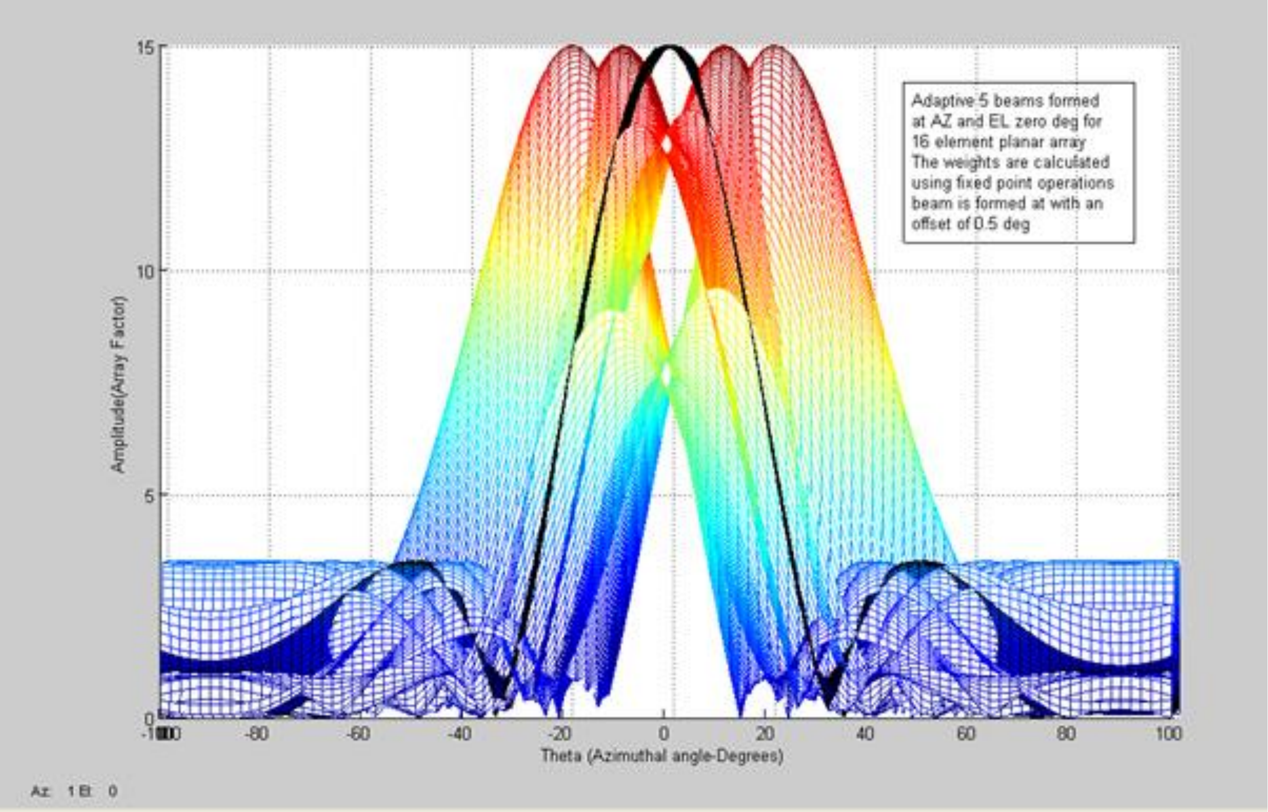


Figure 7-27: Planar array of sixteen elements five adaptive beam.

7.5 Results of Adaptive beams formed with Floating point Operations

Result of implementation of adaptive beam forming using Inverse QRD-RLS algorithm for 16 element planar array is shown in Figure 7-28. Input angle of arrival for desired signal = 0 deg, Interference angle = 35 deg, single beam is produced.

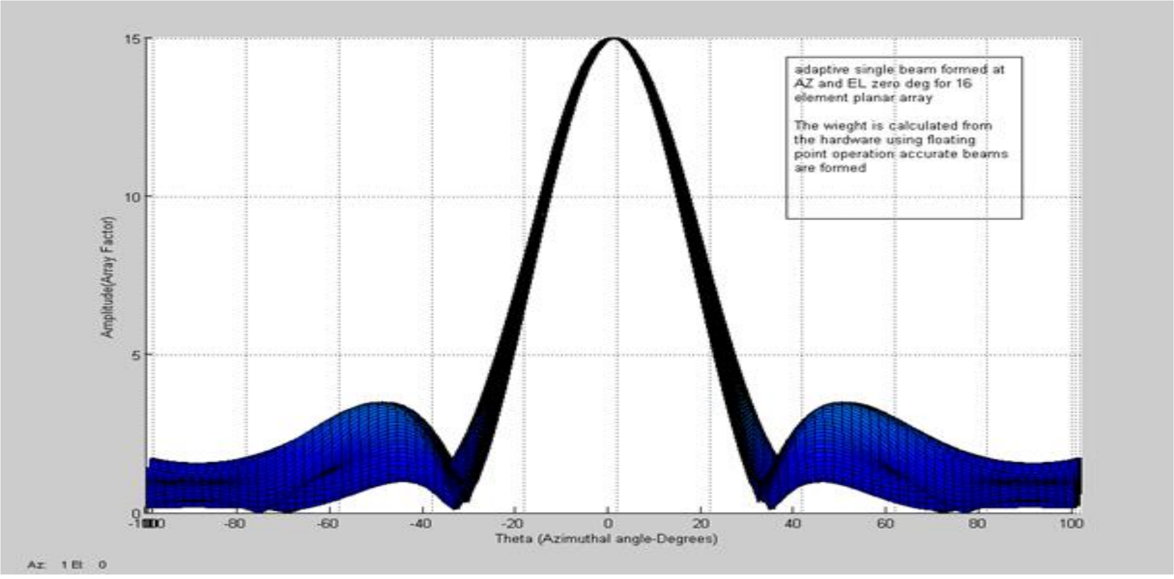


Figure 7-28: Planar array of sixteen elements single adaptive beam.

Result of implementation of adaptive beam forming using Inverse QRD-RLS algorithm for 16 element planar array is shown in Figure 7-29. Input angle of arrival for desired signal = 0 deg, Interference angle =30 deg, 2 beams are generated.

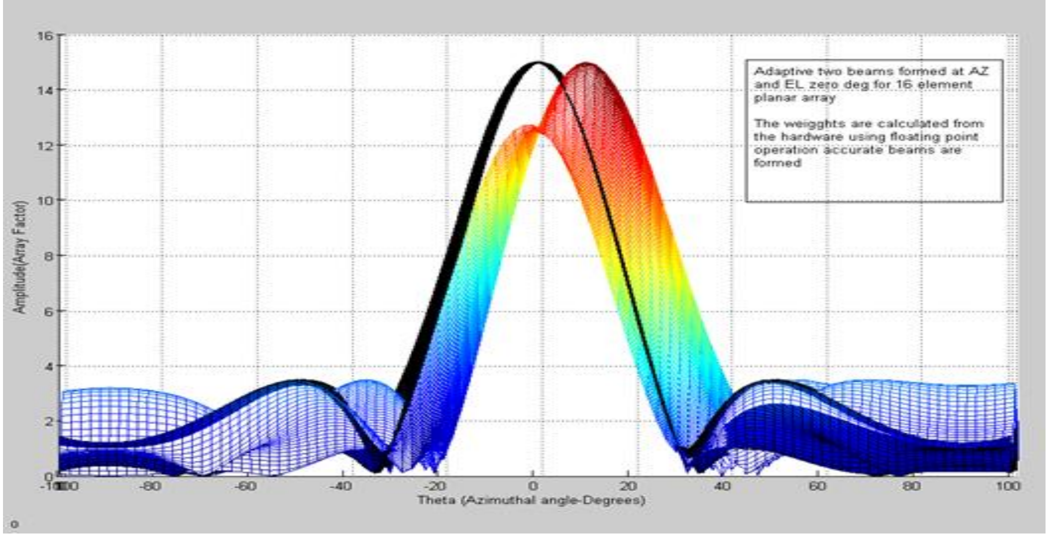


Figure 7-29: Planar array of sixteen elements two adaptive beams.

The Result of implementation of adaptive beam forming using Inverse QRD-RLS algorithm for 16 element planar array is shown in Figure 7-30. The angle of arrival for desired signal = 0 deg, and Interference angle =30 deg, five beams are produced.

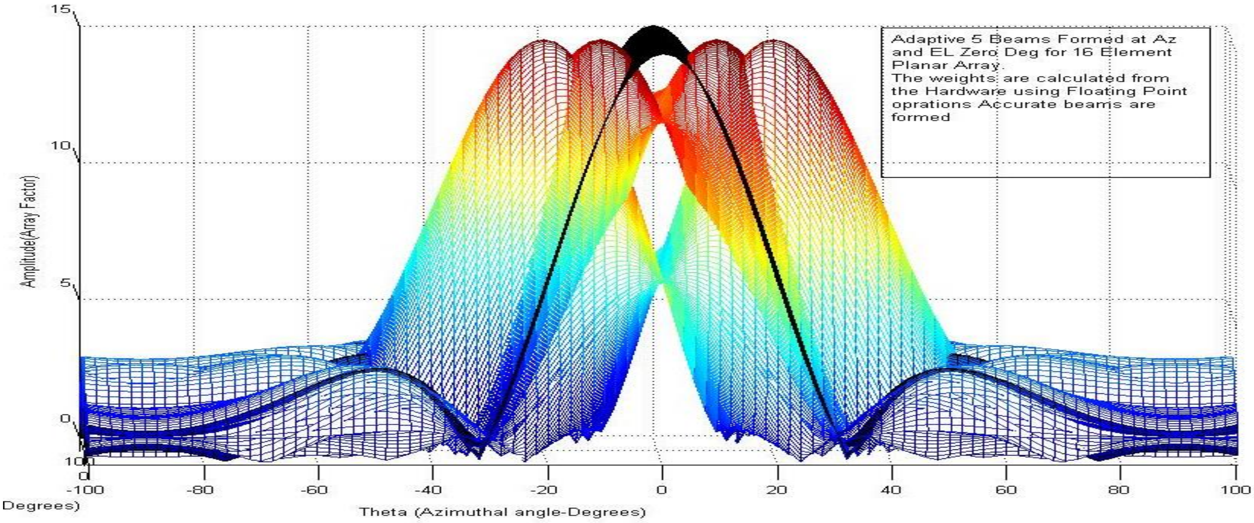


Figure 7-30: Planar array of sixteen elements five adaptive beams.

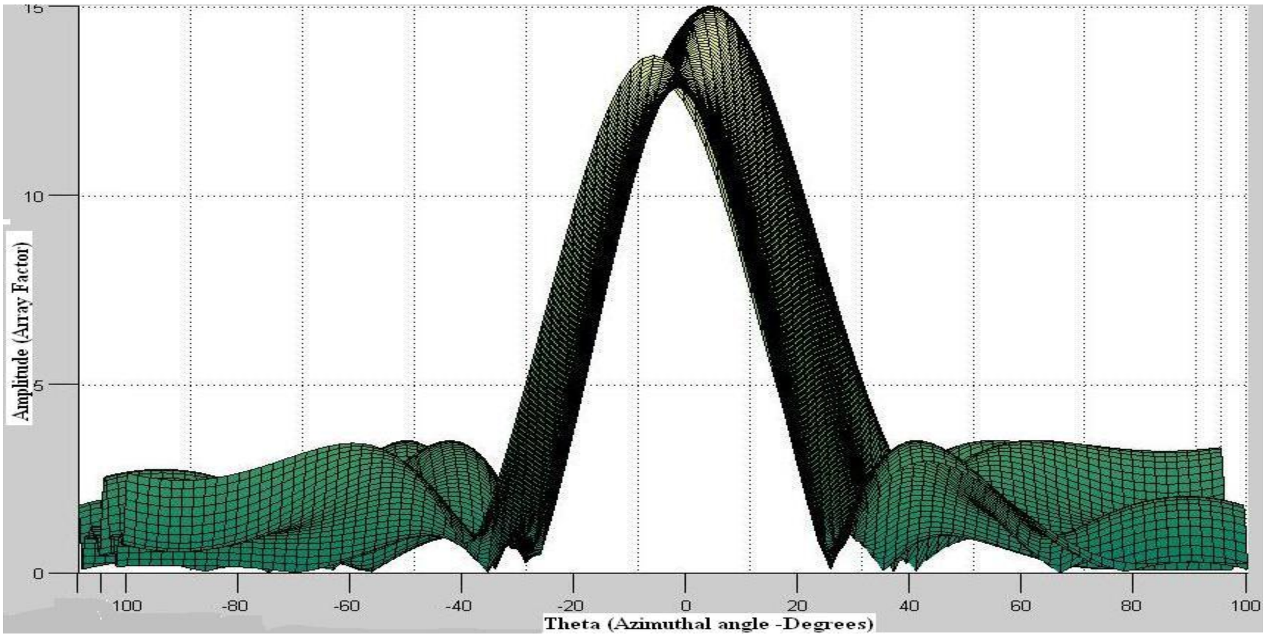


Figure 7-31: Planar array of sixteen elements single beam.

The Result of the implementation of adaptive beam generation with Inverse QRD-RLS algorithm for 16 element planar array is shown in Figure.7-32. Azimuth angle of arrival for desired signal = 15 deg, Elevation angle of arrival for desired signal =45 deg, and four beams are produced.

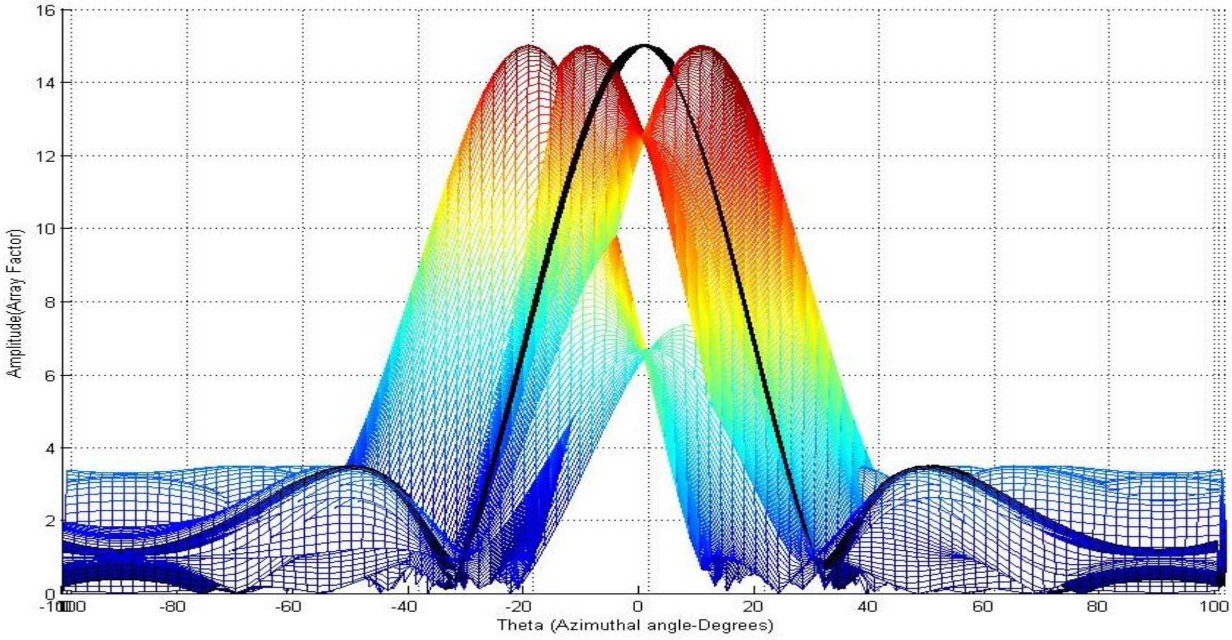


Figure 7-32: Planar array of sixteen elements four adaptive beams.

7.6 Fixed and Floating point Error Analysis

Employing recursive least square algorithm as the base, QR-Decomposition algorithm is designed, due to the unique features like easy mapping of values to the systolic array and the better statistical properties. It is best suited for adaptive beam formation using VLSI method. The main considerations are given to the infinite precision and finite precision point arithmetic representation which is indeed in practice.

The recursive equations in the internal cell limit the speed of the decomposition algorithm of the adaptive filter process. The major factor that must be concentrated during the implementation of the self-modifying digital filter is the quantization noise accumulation. The total number of bits that are present in the digital filter symbolizes the signals which are in turn dependent on the running cost, thus making pathway for the usage of smaller amount of bits to represent a word without distressing the algorithm performance. There by motivating for the error analysis of the fixed point arithmetic representation.

Fixed precision arithmetic is very important because it leads it quicker, smaller efficient units, however it will lead to the outcomes that are less precise, if it’s not rigorously designed. Here it is considered that the tradeoffs applicable for fixed precision illustration that offers the outcomes which are same as an infinite precision implementation. For the fixed point representation the figure of merit is coined as the divergence which arises due to the estimation error.

7.6.1 Analysis of error for QR-Decomposition

During arithmetic calculations, a result might need additional bits and may not be compatible with the reserved bits hence during this instance it is essential to be rigorously handled such that there is no overflow and no erroneous results. In order to stop overflow, each entry within the given matrix should be normalized at the beginning of the decomposition. Normalizing is done by the division of each entry with the biggest matrix entry. Henceforth, making the given matrix entries continually within the range of [0, 1]. Normalization permits for the calculation of the maximum bits which are essential for the large numeric value of the integer and these bits are then reserved to ensure that there is no occurrence of overflow In order to figure out the QRD results for the analysis of the error the comparison is made with respect to the back substitution approach.

The analysis of the error is done by comparing the hardware simulation results with the actual outcome. ISIM software is employed for the implementation the hardware system which simulates our expected design whereas the actual outcome is obtained by employing the design in the MATLAB software. Analysis of the error is completed for the matrix with different entirely three various are utilized for the analysis of error which are termed as mean error, deviation of the error and lastly the percentage error.

7.7 Summary

In this chapter all the experimental results are covered in detail. The adaptive beam former architecture designed and simulated in the hardware simulators and these results are explained. The architecture further realized with the target FPGA device and actual results are captured and explained. The real time data is taken and plotted in a simulator and found that all the results are as expected.

In the next chapter the conclusions and future scope is covered.

Chapter 8

Conclusions and Future Scope

Chapter 8.

Conclusions and Future Scope

8.1 Conclusions

This thesis brings out the design and realization of generic, modular, and scalable adaptive beam former architecture for phased array radar. The architecture development has been carried out in multiple phases. A detailed survey and study has been carried out to analyze various researchers work and limitations of those work. Most of the researchers have emphasized on small section of the architecture developed in this thesis work. A generic architecture suitable for phased array application is not developed by the researchers.

In the first phase of architecture development, the VLSI architecture is designed for realizing the sixteen element array consisting of sixteen digital down convertor modules, complex multipliers and complex adders. The offline optimal weight vectors are computed and stored in the memory of the VIRTEX-V FPGA. Suitable weights were applied and 1/2/4 numbers of beams were generated. Multiple receive beams are formed in the direction of arrival as the radar knows the direction of arrival by putting the transmit beam in that direction. To form one digital beam in the space for sixteen element array sixteen weight vectors are required and on the same line to form N beams from M array, M different weight vectors are required.

In the second phase of architecture development, the previously designed architecture in first phase is modified and improved to form multiple receive beams and configured using VIRTEX-VI FPGA as the requirement of resources have increased. Using the offline calculated weights 6/9 receive beams were generated at simulation level and the same has been validated on the prototype hardware. Traditional method of realization of beam forming is quite complex in nature and more prone to variations against temperature and other environmental conditions. Whereas this VLSI architecture implementation supports modern radars as most of the limitations were overcome by using digital systems. The digital systems are reconfigurable and as per the requirement without going for hardware redesign through programme hardware can be modified as per the system configuration.

During the radar search operation nine beams can be formed to cover the given volume in much less time and during dedicated track or full track of radar operation multiple receive beams can be configured to meet the radar high resolution and greater accuracy requirement.

Further for making beam former system robust and efficient, weights are computed online as per the changing environment in the electronic warfare scenario. The timing requirement for calculating the adaptive weights is within five micro second from the time the sample is arrived from the antenna element. In the final phase of architecture development, the VLSI architecture is redesigned and divided the complete architecture into three FPGA's to calculate the adaptive weights online and to form adaptive multiple receive digital beams. The weights vectors were computed adaptively using the IQRD-RLS algorithm. The modular architecture is implemented on a customized multiple FPGA hardware containing three KINTEX-VII FPGAs. The input data comes to the first two FPGAs from the antenna element and weights are calculated adaptively in FPGA1 and FPGA2, and then multiplied with the input signal using complex multiplier, and then passes the data to the FPGA3. Both FPGA's run synchronously, under the control of the FPGA3, and then the outputs are taken from FPGA3 to the computer using a RS232 interface as well as Chipscope utility. The architecture of FPGA1 is designed to receive the eight element data which will do the down conversion and optimal weight calculation and partial beam formation of eight elements. Similarly FPGA2 will compute for another eight elements. The computation time for adaptive beam formation achieved is 4.2 micro second against the requirement of five micro second. This is the first of its kind architecture developed where adaptive beam formation is possible for futuristic radars. This architecture is validated on the hardware using fixed point arithmetic and floating point arithmetic.

8.2 Future Scope

Modern radar needs FPGA based digital systems for making the system more immune and robust compared to existing analog systems. The adaptive digital beam former system developed can run at a frequency of 100 MHz, depending on the availability of the resources and the input sample rate. The architecture can be modified to meet the higher sampling rates so that it can operate at higher frequencies. The adaptive beam former architecture developed is modular and scalable to any large phased array radar requirement. During the development of this architecture it is assumed that, the ADC's are sampling at 50 MSPS and the Intermediate Frequency (IF) signal is at 60 MHZ received from the antenna element. Due to advancement in the technology in future the ADC's can sample the RF signal of the order of few GHz directly and give the digital data which can be processed to generate the optimal weights and to form the adaptive beams. Since the architecture is modular only the front modules needs to be upgraded to handle the high speed data coming from the antenna element through the ADCs. The current architecture is designed for sixteen element phased array configuration and the same can be extended to larger array size of 32 elements as the architecture is modular. Hence the architecture can be modified to meet the future phased array radar requirement at much higher sampling rates. The architecture developed is most suitable for phased array radar application and can be extended for planar arrays of larger array dimensions. The adaptive beam former architecture can be extended for radars operating in other bands without much modification. The designed VLSI architecture can be converted to an ASIC, which will reduce the board size and considerably reduce the cost of the phased array system.

Publications

- **International Journals**

1. D. GovindRao, T. Kishore Kumar, N. S. Murthy And A.Vengadarajan, "Design and Realization of Array Signal Processor VLSI Architecture for Phased Array System", American Journal of Engineering Research (AJER), e-ISSN: 2320-0847, p-ISSN : 2320-0936, Volume-5, Issue-7, July-2016, pp-253-261.
2. D. GovindRao, T. Kishore Kumar, N. S. Murthy And A.Vengadarajan, "Novel Method of Realization of Scalable VLSI Adaptive Digital Beamforming Architecture for Phased Array Radar", International Journal of Engineering Research and Development, e-ISSN: 2278-067X, p-ISSN: 2278-800X, Volume 12, Issue 7, July-2016, PP-27-35.
3. D GovindRao, N S Murthy, A,Vengadarajan "Adaptive VLSI Architecture of Beam Former for Active Phased Array Radar" International Journal of New Computer Architectures and their Applications , IJNCAA, Volume 3, Issue 2, Sep-2013, PP-19-29.

- **International Conferences**

1. D. GovindRao, N.S.Murthy, A.Vengadarajan, "Design and Implementation of Digital Beam Former Architecture for Phased Array Radar", International Radar Symposium India, PP, 218-223, 2013.
2. D. GovindRao, A.Vengadarajan, N.S.Murthy "Digital Beamforming architecture for sixteen element planar phased array radar" TAEECE, IEEE, pp 532-537, May 2013.
3. D. GovindRao, N. S. Murthy, A.Vengadarajan "Efficient filter implementation using QRD-RLS algorithm for phased array radar application" TAEECE, IEEE, pp 224-229, May 2013.

References

[1]. Givens, W. 1958. Computation of plane unitary rotations transforming a general matrix to triangular form. In J. Soc. Indust. Appl. Math. 6, 26—50, 1958.

[2]. J. Volder, “The CORDIC Trigonometric Computing Technique,” IRE Transactions on Electronic Computers, vol. EC-8, no. 3, pp. 330–334, 1959.

[3]. W. H. Gentleman and H. T. Kung, “Matrix triangularization by systolic arrays”. SPIE Real-Time Signal Processing IV, vol. 298, pp. 19–26, January 1981.

[4]. H. K. W.M. Gentleman, “Matrix Triangularization by Systolic Arrays,” Real-Time Signal Processing, vol. 298, pp. 19–26, 1981.

[5]. J. G.McWhirter, “Recursive least-squares minimization using a systolic array”. SPIE Real-Time Signal Processing VI, vol. 431, pp. 105–112, January 1983.

[6]. G. Carayannis, D. Manolakis, and N. Kalouptsidis, “A unified view of parametric processing algorithms for prewindowed signals,” Signal Process., vol. 10, no. 4, pp. 335–368, Jun. 1986.

[7]. Christopher R. Ward, Philip J. Hargrave, Ant John G. McWhirter, “A Novel Algorithm and Architecture For Adaptive Digital Beamforming”, IEEE Transactions On Antennas And Propagation, Pp. 338-346, IEEE-1986.

[8]. D. Psaltis, A. Sideris, and A. A. Yamamura, “A multilayered neural network controller,” IEEE Control Syst. Mag., vol. 8, pp. 17–21, Apr. 1988.

[9]. J. R. Cavallaro and F. Luk, “CORDIC Arithmetic for an SVD Processor,” in Journal of Parallel and Distributed Computing, June 1988.

[10]. D. T. M. Slock, “Reconciling fast RLS lattice and QR algorithms,” in Proc. Int. Conf. Acoust., Speech, Signal Process. ICASSP’90, Albuquerque, NM, Apr. 1990, vol. 3, pp. 1591–1594.

[11]. K. H. Kim and E. J. Powers, “Analysis of initialization and numerical instability of fast RLS algorithms,” in Proc. Int. Conf. Acoust., Speech, Signal Process. ICASSP’91, Toronto, Canada, Apr. 1991.

[12]. P. A. Regalia and M. G. Bellanger, “On the duality between fast QR methods and lattice methods in least squares adaptive filtering,” IEEE Trans. Signal Process., vol. 39, no. 4, pp. 876–891, Apr. 1991

[13]. S. Haykin, Adaptive Filter Theory. Prentice Hall, third ed. R. Dohler, “Squared Givens Rotation,” IMA Journal of Numerical Analysis, no. 11, pp. 1–5, 1991.

[14]. J.Gotze and U.Schwiegelshohn, “A Square Root and Division Free Givens Rotation for Solving Least Square Problems on Systolic Arrays,” J. SCI. STAT. COMPUT., vol. 12, pp. 800–807, July 1991.

[15]. S. T. Alexander and A. L. Ghirnikar, “A method for recursive least squares filtering based upon an inverse QR decomposition,” IEEE Trans. Signal Process, vol. 41, no. 1, pp. 20–30, Jan. 1993.

[16]. T. Tanaka, I. Chiba, R. Mtura and Y. Karasawa, “Digital Signal Processor for Digital Multi Beam Forming Antenna in Mobile Communication,” ATR Optical and Radio Communications Research Laboratories, Japan, pp. 1507-1511, 1994.

[17]. K. Kota and J. Cavallaro, “Numerical Accuracy and Hardware Tradeoffs for CORDIC Arithmetic for Special-Purpose Processors,” in IEEE Transactions on Computers, vol. 42, pp. 769–779, July 1993.

[18]. M. A. Syed and V. J. Mathews, “QR-decomposition based algorithms for adaptive Volterrafiltering,” IEEE Trans. Circuits Syst. I: Fund. Theory Appl., vol. 40, no. 6, pp. 372–382, Jun. 1993.

[19]. N. Hemkumar and J. Cavallaro, “Redundant and Online CORDIC for Unitary Transformations,” in IEEE Transactions on Computers, vol. 43, pp. 941–954, August 1994.

[20]. A. Björck, “Numerics of Gram-Schmidt Orthogonalization,” Linear Algebra and Its Applications, 198:297-316, 1994.

[21]. G.H. Golub and C. F. Van Loan, “Matrix Computations,” 3 rd ed. John Hopkins University Press, Baltimore, MD, 1996.

[22]. S. Haykin, Adaptive Filter Theory, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[23]. C. Eun and E. J. Powers, “A new Volterrapredistorter based on the indirect learning architecture,” IEEE Trans. Signal Process. vol. 45, no. 1, pp. 20–30, Jan. 1997.

[24]. J. A. Apolinário, Jr. and P. S. R. Diniz, “A new fast QR algorithm based on a priori errors,” IEEE Signal Process. Lett., vol. 4, no. 11, pp. 307–309, Nov. 1997.

[25]. Ray Andraka, “A Survey of CORDIC Algorithms for FPGA based computers,” International Symposium on Field Programmable Gate Arrays, 1998.

[26]. Lightbody, G.; Woods, R.; McCanny, J.; Walke, R.; Hu, Y.; Trainor, D., “Rapid design of a single chip adaptive beamformer,” IEEE Workshop on Signal Processing Systems, 1998.

[27]. A. A. Rontogiannis and S. Theodoridis, “Multichannel fast QRD-LS adaptive filtering: New technique and algorithms,” IEEE Trans. Signal Process., vol. 46, no. 11, pp. 2862–2876, Nov. 1998.

[28]. R. L. Walke, R.W.M. Smith and G. Light boy, “Architectures for adaptive weight calculation on ASIC and FPGA”, Proc. 33rd Asilomar Conference on Signals, Systems and Computers, pp 1375-1380, 1999.

[29]. Lightbody, G.; Walke, R.; Woods, R.; McCanny, J., “Parameterisable QR core,” Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers, 1999.

[30]. R. Walke, R. Smith, and G.Lightbody, “Architectures for Adaptive Weight Calculation on ASIC and FPGA,” in Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers, vol. 2, pp. 1375 – 1380, 24-27 Oct 1999.

[31]. Jun Ma; Parhi, K.K.; Deprettere E.F., "Annihilation-reordering lookahead pipelined CORDIC-based RLS adaptive filters and their application to adaptive beamforming,” IEEE Transactions on Signal, Processing, 2000.

[32]. Alan J. Fenn, Donald H. Temme, William P. Delaney, and William E. Courtney, “The Development of Phased-Array Radar Technology”,Volume 12, Number 2,Lincoln Laboratory Journal, pp. 321-340, 2000.

[33]. C. A. Medina, J. Apolinário, Jr., and M. Siqueira, “A unified framework for multichannel fast QRD-LS adaptive filters based on backward prediction errors,” in Proc. IEEE Int. Midwest Symp. Circuits Syst. MWSCAS’O2, Tulsa, OK, Aug. 2002.

[34]. M. Bouchard, “Numerically stable fast convergence least-squares algorithms for multichannel active sound cancellation systems and sound deconvolution systems,” Signal Process., vol. 82, no. 5, pp. 721–736, May 2002.

[35]. J. A. Apolinário, Jr., M. G. Siqueira, and P. S. R. Diniz, “Fast QR algorithms based on backward prediction errors: A new implementation and its finite precision performance,” Circuits Syst. Signal Process., vol. 22, pp. 335–349, Jul./Aug. 2003.

[36]. Lightbody, G.; Woods, R.; Walke, R., “Design of a parameterizable silicon intellectual property core for QR-based RLS filtering,” IEEE Transactions on Very Large Scale Integration ,VLSI Systems, 2003.

[37]. Zhaohui Liu; McCanny, J.V.; Lightbody, G.; Walke, R., Generic SoC "QR array processor for adaptive beamforming," IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 2003.

[38]. J. Yue, K. J. Kim, J. Gibson, and R.A.Iltis, "Channel estimation and data detection for MIMO-OFDM systems," in In Proceedings of IEEE Global Telecommunications Conference, vol. 2, pp. 581 – 585, 1-5 Dec 2003.

[39]. A. Ramos, J. Apolinário, Jr., and M. G. Siqueira, "A new order recursive multiple order multichannel fast QRD algorithm," in Proc. 38th Ann. Asilomar Conf. Signals, Syst., Comput., Pacific Grove, CA, Nov. 2004.

[40]. Deepak Boppana, KullyDhanoa, Jesse Kempa, "FPGA based Embedded Processing Architecture for the QRD-RLS Algorithm", IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'04), 2004.

[41]. Antonio L. L. Ramos , Jose A. Apolin´ario Jr. and Marcio G. Siqueira, "A New Order Recursive Multiple order Multichannel Fast QRD-RLS Algorithm", IEEE Global Telecommunications Conference, pp. 965-969, 2004.

[42]. Deepak Boppana , Kully Dhanoa , Jesse Kempa, "FPGA based Embedded Processing Architecture for the QRD-RLS Algorithm", Proceedings of the 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines,2004.

[43]. Boppana, D.; Dhanoa, K.; Kempa, J., "FPGA based embedded processing architecture for the QRD-RLS algorithm," 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, 2004. FCCM 2004.

[44]. L. Ding, G. T. Zhou, D. R. Morgan, Z. Ma, J. S. Kenney, J. Kim, and C. R. Giardina, "A robust digital baseband predistorter constructed using memory polynomials," IEEE Trans. Common., vol. 52, no. 1, pp. 159–165, Jan. 2004.

[45]. A. Ramos and J. A. Apolinário, Jr., "A newmultiple order multichannel fast QRD algorithm and its application to non-linear system identification," in Proc. XXI Simp. Brasileiro de Telecomun. SBT'04, Belém, PA, Brazil, Sep. 2004.

[46]. Karkooti, M.; Cavallaro, J.R.; Dick, C., "FPGA Implementation of Matrix Inversion Using QRD-RLS Algorithm," Thirty-Ninth Asilomar Conference on Signals, Systems and Computers, 2005.

[47]. M. Karkooti, J. R. Cavallaro, and C. Dick, "FPGA implementation of matrix inversion using QRD-RLS algorithm," in Proceedings of the 39th Asilomar Conference on Signals, Systems and Computers, pp. 1625–1629,November 2005.

[48]. H. Yang, "A road to future broadband wireless access: MIMO-OFDM Based air interface," IEEE Communications Magazine, vol. 43, pp. 53– 60, Jan 2005.

[49]. M. Myllyla, J. Hintikka, J. Cavallaro, M. Juntti, M. Limingoja, and A. Byman, "Complexity Analysis of MMSE Detector Architecture for MIMO OFDM Systems," in Proceedings of the 2005 Asilomar conference,Pacific Grove, CA, Oct 30 - Nov 2 2005.

[50]. MarjanKarkooti, Joseph R. Cavallaro and Chris Dick, "FPGA Implementation of Matrix Inversion UsingQRD-RLS Algorithm, IEEE-05, pp. 1625-1629, 2005.

[51]. Dick, Chris; Harris, Fred; Pajic, Miroslav; Vuletic, Dragan; "Real-TimeQRD-Based Beamforming on an FPGA Platform," Fortieth Asilomar Conference on Signals, Systems and Computers, ACSSC,2006.

[52]. Deschamps, J. P., Bioul G. J. A., Sutter G., "Synthesis of ArithmeticCircuits," FPGA, ASIC and Embedded Sytems. John Wiley & Sons, Inc., 2006.

[53]. M. D. Vahey, J. J. Granacki, L. J. Lewins, D. Davidoff, J. T. Draper, C. S.Steele, G. K. Groves, M. Kramer, J. LaCoss, K. Prager, J. Kulp, and C. Channell, "MONARCH: A first generation polymorphic computing processor," in 10th High Performance Embedded Computing Workshop, Sep. 2006

[54]. F. Sobhanmanesh and S. Nooshabadi, "Parametric minimum hardware QR-factoriser architecture for V-BLAST detection," IEE Proceedings: Circuits, Devices and Systems, vol. 153, no. 5, pp. 433–441, 2006.

[55]. C. K. Singh, S. H. Prasad, and P. T. Balsara, "A fixed-point implementation for QR decomposition," in Proceedings of the IEEE Dallas ICAS Workshop on Design, Applications, Integration and Software, pp. 75–78, Richardson, Tex, USA, October 2006.

[56]. M. Shoaib, S. Werner, J. A. Apolinário, Jr., and T. I. Laakso, "Equivalent output-filtering using fast QRD-RLS algorithm for burst-type training applications," in Proc. ISCAS'2006, Kos, Greece, May 2006.

[57]. D. R. Morgan, Z. Ma, J. Kim, M. G. Zierdt, and J. Pastalan, "A generalized memory polynomial model for digital predistortion of RF power amplifiers," IEEE Trans. Signal Process, vol. 54, no. 10, pp. 3852–3860, Oct. 2006.

[58]. M. Shoaib, S.Werner, J. A. Apolinário, Jr., and T. I. Laakso, "Solution to the weight extraction problem in FQRD-RLS algorithms," in Proc. Int. Conf. Acoust., Speech, Signal Process. ICASSP'06, Toulouse, France, May 2006.

[59]. M. Shoaib, S. Werner, J. A. Apolinário, Jr., and T. I. Laakso, "Multichannel fast QR-decomposition RLS algorithms with explicit weight extraction," in Proc. EUSIPCO'2006, Florence, Italy, Sep. 2006.

[60]. MobienShoaib,Stefan Werner,Jos´e A. Apolin´ario Jr.and Timo I. Laakso, "Solution to the Weight Extraction Problem in Fast QR-Decomposition RLS Algorithms", IEEE-06, pp. 572- 575, 2006.

[61]. O. Tamer, "FPGA based smart antenna implementation", PhD report, Graduate School of Natural and Applied Sciences of DokuzEylül University, pp. 6-20, 2007.

[62]. Singh, C.K.; Sushma Honnavara Prasad; Balsara, P.T., "VLSI Architecture for Matrix Inversion using Modified Gram-Schmidt based QR Decomposition," 20th International Conference on VLSI Design, 2007.

[63]. C. K. Singh, S. H. Prasad, and P. T. Balsara, "VLSI architecture for matrix inversion using modified gram-schmidt based QR decomposition," Proceedings of the 20th International Conference on VLSI Design Held Jointly with 6th International Conference on Embedded Systems, pp. 836–841, 2007.

[64]. K. K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation, John Wiley & Sons, 2007.

[65]. A. Ramos, J. A. Apolinário, Jr., and S. Werner, "Multichannel fast QRD-RLS adaptive filtering: Block-channel and sequential algorithm based on updating backward prediction errors," Signal Process., vol. 87, pp. 1781–1798, Jul. 2007.

[66]. M. Shoaib, S. Werner, and J. A. Apolinário Jr., "Reduced complexity solution for weight extraction in QRD-LSL algorithm," IEEE Signal Process. Lett., vol. 15, pp. 277–280, 2008.

[67]. P. S. R. Diniz, Adaptive Filtering: Algorithms and Practical Implementation, 3rd Ed. New York: Springer, 2008.

[68]. D. Theodoropoulos, G. Kuzmanov, and G. Gaydadjiev, "A reconfigurable beamformer for audio applications," in IEEE 7th Symposium on Application Specific Processors (SASP '09). Los Alamitos, CA, USA: IEEE Computer Society, Jul. 2009, pp. 80–87.

[69]. J. Antonio Apolin´ario Jr., QRD-RLS Adaptive Filtering, 1st ed. Springer Publications, pp 283-291, 2009.

[70]. X. Wang, M. Leeser, “A truly two-dimensional systolic array FPGA implementation of QR Decomposition”. ACM Transactions on Embedded Computing Systems, Vol. 9 No. 1, Article 3, pp 1-10. October 2009.

[71]. D. Patel, M. Shabany, and P. G. Gulak, “A low-complexity high speed QR decomposition implementation for MIMO receivers,”in Proceedings of the IEEE International Symposium on Circuits and Systems ,ISCAS ’09, pp. 33–36, Taipei, Taiwan, May 2009.

[72]. Yan Ye, Taijun Liu, KunshengRen and XingbinZeng, “Modified Systolic Array Implementation of QRD-RLS Algorithms for Solving the Generalized Hammerstein Models”, Proceedings of the 15th Asia-Pacific Conference on Communications (APCC 2009), pp.213-216, 2009.

[73]. Dimpesh Patel, Mahdi Shabany and P. Glenn Gulak, “A Low-Complexity High-Speed QR DecompositionImplementation for MIMO Receivers”, IEEE-09, pp. 33-36, 2009.

[74]. J. A. Apolinário Jr., QRD-RLS Adaptive Filtering. New York: Springer, 2009.

[75]. W. H. Wedon, “Phased array digital beam forming hardware development at applied radar” IEEE international symposium on Phased array systems and technology, pp-854-859, 2010.

[76]. MobienShoaibandJosé Antonio Apolinário, “Multichannel Fast QR-Decomposition Algorithms:Weight Extraction Method and Its Applications”, IEEE Transactions on Signal Processing, pp.175-188, IEEE – Jan. 2010.

[77]. S.Haykin, “Adaptive Filter Theory,” 4th Edition, Pearson, ISBN 978-81-317-0869-9, pp 4-22, 2011.

[78]. D. Chen and M. Sima, “Fixed-point CORDIC-based QR decomposition by givens rotations on FPGA,” in Proceedings of the International Conference on Reconfigurable Computing and FPGAs ,ReConFig ’11, pp. 327–332, IEEE, Canc´un,Mexico, December 2011.

[79]. Mohammad Salman Baig, B. RamaswamyKarthikeyan, DipayanMazumdar and Govind R. Kadambi, “Improved Receiver Architecture for Digital Beamforming Systems” International Conference on Computer, Communication and Electrical Technology-IEEE, pp. 208-214, 2011.

[80]. Dongdong Chen and Mihai SIMA, “Fixed-Point CORDIC-Based QR Decomposition byGivens Rotations on FPGA”, International Conference on Reconfigurable Computing and FPGAs, pp.327-332, IEEE – 2011.

[81]. Cong Xiang, Da-ZhengFeng, HuiLv,JieHe, and Hong-WeiLiu, “Three-dimensional reduced-dimension transformation for MIMO radar space–time adaptive processing” journal of Elsevier – signal processing, pp. 2121–2126, 2011.

[82]. SumitVerma, ArvindPathak, “Digital beam forming using RLS – QRD algorithm”, International Journal of Engineering Research & Technology, Vol. 1 Issue 5, and ISSN: 2278-0181, 2012.

[83]. SufengNiu, SemihAslan and JafarSaniie, “FPGA Implementation of Fast QR DecompositionBased on Givens Rotation”, IEEE-12, pp. 470-473, 2012.

[84]. Xilinx, http://www.xilinx.com, Kintex-VII, FPGA Data Sheets and Application, Xilinx Inc.2012.

[85]. S. Aslan, S. Niu, and J. Saniie, “FPGA implementation of fast QR decomposition based on givens rotation,” in Proceedings of the IEEE 55th International Midwest Symposium on Circuits and Systems ,MWSCAS ’12, pp. 470–473, IEEE, August 2012.

[86]. S.Niu, S. Aslan, and J. Saniie, “FPGA based architectures for high performance adaptive FIR filter systems,” in Proceedings of the IEEE International Instrumentation and Measurement Technology Conference ,I2MTC ’13, pp. 1662–1665, 2013.

[87]. U. Vishnoi and T. G. Noll, “A family of modular area- and energy-efficient QRD-accelerator architectures,” in Proceedings of the International Symposium on System on Chip, SoC '13, pp.1–8, Tampere, Finland, October 2013.

[88]. B. Han, Z. Yang, and Y. R. Zheng, “FPGA implementation of QR decomposition for MIMO-OFDM using four CORDIC cores,” in Proceedings of the IEEE International Conference on Communications ,ICC ’13, pp. 4556–4560, June 2013.

[89]. J. Rust, F. Ludwig, and S. Paul, “Low complexity QR-decomposition architecture using the logarithmic number system,” in Proceedings of the Conference on Design, Automation and Test in Europe, pp. 97–102, EDA Consortium, 2013.

[90]. I. H. Kurniawan, J.-H. Yoon, and J. Park, “MultidimensionalHouseholder based high-speed QR decomposition architecture for MIMO receivers,” in Proceedings of the IEEE International Symposium on Circuits and Systems ,ISCAS ’13, pp. 2159–2162, Beijing, China, May 2013.

[91]. M. Shabany,D. Patel, and P. G.Gulak, “Alow-latency low-power QR-decomposition ASIC implementation in 0.13 μm CMOS,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol.60, no. 2, pp. 327–340, 2013.

[92]. IputHeriKurniawan, Ji-Hwan Yoon and Jongsun Park, “Multidimensional Householder basedHigh-Speed QR Decomposition Architecture for MIMO Receivers”, IEEE-13, pp. 2159-2162, 2013.

[93]. Xilinx, http://www.xilinx.com, 7 Series FPGAs Overview, DS180 ,v1.14, July 29, 2013

[94]. Bing Han, Zengli Yang, and Yahong Rosa Zheng, “FPGA Implementation of QR Decomposition forMIMO-OFDM Using Four CORDIC Cores”, IEEE Signal Processing for Communications Symposium, pp. 4556-4560, 2013.

[95]. SufengNiu, Sizhou Wang, SemihAslanand JafarSaniie, “Hardware and Software Design for QR-Decomposition Recursive Least Square Algorithm”, IEEE-13, pp. 117-120, 2013.

[96]. MobienShoaiband SalehAlshebeili, “A Fast Widely-Linear QR-Decomposition Least-Squares (FWL-QRD-RLS)Algorithm”, IEEE-13, pp. 4169-4172, 2013.

[97]. Jochen Rust, Frank Ludwig and Steffen Paul, “Low Complexity QR-Decomposition ArchitectureUsing the Logarithmic Number System”, EDAA, 2013.

[98]. UpasnaVishnoi and Tobias G.NolI, “A Family of Modular Area- and Energy-EfficientQRD-Accelerator Architectures”, IEEE, 2013.

[99]. Hyunwook Yang and Seungwon Choi, “Implementation of a Zero-Forcing PrecodingAlgorithm Combined with Adaptive BeamformingBased on WiMAX System” International Journal of Antennas and Propagation, 2013.

[100]. R. Gayathri and J. Sheeba Rani, “Fixed point pipelined architecture for QR decomposition,” in Proceedings of the IEEE International Conference on Advanced Communication Control and Computing Technologies ,ICACCCT ’14, pp. 468–472,IEEE, 2014.

[101]. F. Riera-Palou, “Reconfigurable structures for direct equalization in mobile receivers [Ph.D. thesis],” University of Bradford, Bradford, UK, 2014.

[102]. OwaisTalaatWaheed, AymanShabra, Ibrahim and M. Elfadel, “FPGA Methodology for Power Analysis ofEmbedded Adaptive Beamforming”, IEEE -2015.

[103]. Kanchan H. Wagh, “Microstrip Array Antenna and Beamforming Algorithm for Phased Array Radar”, International Journal of Advanced Research in Education & Technology (IJARET), Vol. 2, Issue 3, pp. 148-151, 2015.

[104]. Hang Hu, “Aspects of the Subarrayed Array Processing forthe Phased Array Radar”, International Journal of Antennas and Propagation, 2015.

[105]. Hongtao Li, Ke Wang, Chaoyu Wang, Yapeng He and Xiaohua Zhu, “Robust Adaptive Beamforming Based on Worst-Caseand Norm Constraint”, International Journal of Antennas and Propagation, 2015.

[106]. KuandongGao, Huaizong Shao, JingyeCai, Hui Chen and Wen-Qin Wang, “Frequency Diverse Array MIMO Radar AdaptiveBeamforming with Range-Dependent Interference Suppressionin Target Localization” International Journal of Antennas and Propagation, 2015.

[107]. Xiaojun Mao, Wenxing Li, Yingsong Li, Yaxiu Sun and ZhuqunZhai, “Robust Adaptive Beamforming against Signal Steering Vector Mismatch and Jammer Motion” International Journal of Antennas and Propagation, 2015.

[108]. Lin Li, Fangfang Chen and Jisheng Dai, “Separate DOD and DOA Estimation for Bistatic MIMO Radar”, International Journal of Antennas and Propagation, 2016.

[109]. Xilinx. XILINX Logic core floatinghttp://www.xilinx.com/bvdocs/ipcenter/datasheet.pdf

[110]. Simon Haykin, “Adaptive Filter Theory”, Prentice Hall, Fourth Edition.

[111]. www.Xilinx.xom

[112]. William L. Melvin and James A. Scheer, “Principles of Modern Radar”, Vol-II, Scitech Publishing, 2013.

[113]. Marcel D. van de Burgwal, Kenneth C. Rovers, Koen C.H. Blom, André B.J. Kokkeler, and Gerard J.M. Smit, “Adaptive Beamforming using the Reconfigurable MONTIUM TP” 13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools, 2010.

[114]. Xiaojun Wang Airvana and Miriam Leeser ”A Truly Two-Dimensional Systolic Array FPGA Implementation of QR Decomposition”, ACM Transactions on Embedded Computing Systems, Vol. 9, No. 1, Article 3, Publication date: October 2009.

[115]. H. J. “Visser, Array and phased array antenna basics. Chichester”, West Sussex, UK: Wiley, Sep. 2005.

[116]. M. I. Skolnik, Introduction to Radar Systems, 3rd ed. New York, NY, USA: McGraw-Hill, 2001.

[117]. Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for 11/12 GHz satellite services, European Telecommunication Standard Institute (ETSI), SophiaAntipolis, France, Aug. 1997, EN 300 421 v1.1.2. [Online]. Available: http://www.etsi.org/deliver/etsi_en/300400_300499/300421/01.01.02_60/en_300421v010102p.pdf

[118]. B. H. Allen and M. Ghavami, Adaptive Array Systems, Fundamentals and Applications. Chichester, West Sussex, UK: Wiley, 2005.

[119]. Marcel D. van de Burgwal, Kenneth C. Rovers, Koen C.H. Blom, André B.J. Kokkeler, and Gerard J.M. Smit, “Adaptive Beamforming using the Reconfigurable MONTIUM TP” Faculty of Electrical Engineering, Math and Computer Science,University of Twente Enschede, The Netherlands

[120]. R. H. Roy, A. J. Paulraj, and T. Kailath, “Esprit – a subspace rotation approach to estimation of parameters of cisoids in noise,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 34, no. 5, pp. 1340–1342, Oct. 1986.

[121]. R. O. Schmidt, “Multiple emitter location and signal parameter estimation,” IEEE Transactions on Antennas and Propagation, vol. AP-34, no. 3, pp. 276–280, Mar. 1986.

[122]. J. R. Treichler and B. G. Agee, “A new approach to multipath correction of constant modulus signals,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 31, no. 2, pp. 459–472, Apr. 1983.

[123]. Z. Xu, “New cost function for blind estimation of M-PSK signals,” in IEEE Wireless Communications and Networking Conference, vol. 3, Sep. 2000, pp. 1501–1505.

[124]. K. C. Rovers, M. D. van de Burgwal, A. B. J. Kokkeler, and G. J. M. Smit, “Rationale for and design of a generic tiled hierarchical phased array beamforming architecture,” in Proc. 18th Annual Workshop on Circuits, Systems and Signal Processing (ProRISC 2007). Utrecht, The Netherlands: Technology Foundation, Nov. 2007, pp. 160–168.

[125]. P. M. Heysters, “Coarse-grained reconfigurable processors – flexibility meets efficiency,” Ph.D. dissertation, University of Twente, Enschede, The Netherlands, Sep. 2004.

[126]. G. K. Rauwerda, P. M. Heysters, and G. J. M. Smit, “Towards software defined radios using coarse-grained reconfigurable hardware,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 16, no. 1, pp. 3–13, Jan. 2008.

[127]. G. J. M. Smit, A. B. J. Kokkeler, P. T. Wolkotte, P. K. F. Hölzenspies, M. D. van de Burgwal, and P. M. Heysters, “The Chameleon architecture for streaming DSP applications,” EURASIP Journal on Embedded Systems, vol. 2007, no. 78082, Jan. 2007.

[128]. K. C. H. Blom, “DVB-S signal tracking techniques for mobile phased arrays,” Master’s thesis, University of Twente, Enschede, The Netherlands, Dec. 2009

[129]. Sasan Houston Ardalan and S. T. Alexander “Fixed-point Roundoff Error Analysis of the Exponentially Windowed RLS Algorithm for Time-Varvinrr Svsterns” IEEE transactions on acoustics, speech, and signal processing, vol. Assp-35, no. 6, June 1987.