Adaptive VLSI Architecture of Beam Former for Active Phased Array Radar

D. Govind Rao\textsuperscript{1}, N. S. Murthy\textsuperscript{2} and A.Vengadarajan\textsuperscript{3}
\textsuperscript{1,3}LRDE, DRDO, Bangalore
\textsuperscript{2}NIT, Warangal
E-mail: dgrao@lrde.drdo.in, nsm@nitw.ac.in

ABSTRACT

This paper describes architecture for a digital beamformer developed for 16 element phased array radar. The digital beamformer architecture includes the complex operations such as down conversion which is done in parallel for the signal coming from each of the antenna elements and the filtering. A high performance FPGA is employed to perform these operations. An echo signal of 5 MHz riding on the IF signal of 60 MHz is down converted digitally to the baseband of the echo signal. The baseband echo signal is then multiplied by the complex weights and then summed to form the beam. The complex weights can be computed offline and online using two different approaches for implementation of an adaptive filter using QRD-RLS (QR decomposition based Recursive Least Squares) algorithm for Phased Array Radar application. One approach involves back substitution procedure whereas another involves updating of inverse data matrix. The former approach is called as Conventional QRD-RLS and the later is called as Inverse QRD-RLS. The developed VLSI Architecture employs 16 bit 125 MS/s ADCs and a very high performance state of the art Xilinx FPGA device Vertex-IV VLX240T to form multiple receive beams simultaneously. The device used has large number of on chip resources for the parallel processing and the 200MHz clock generator. The complex weights are externally calculated OR internally calculated adaptively using highly stable Q-R decomposition and Inverse Q-R decomposition based recursive least squares algorithm.

KEYWORDS

Virtex, QRD-RLS, DDC, CIC filter, SOC, Complex Arithmetic.

1. INTRODUCTION

Systems such as Radar receiver [9] which are designed to receive spatially propagating signals often encounter the presence of interference signals. If the desired signal and interference occupy the same frequency band, then temporal filtering cannot be used to separate signal from interference. However, the desired and interfering signal usually originate from different spatial locations. This spatial separation can be exploited to separate signal from interference using spatial filter at the receiver. When the spatial sampling is discrete, the hardware that performs spatial filtering is termed as beam former.

Digital beam forming [13] has many of the advantages a digital computational environment has over its analog counterpart. In most cases, less power is needed to perform the beam steering of the phased array antenna [11]. Another advantage is the reduction of variations associated with time, temperature, and other environmental changes found in analog devices. Digital beam formers can accomplish minimization of side-lobe levels, interference cancelling and
multiple beam operation without changing the physical architecture of the phased array antenna. Every mode of operation of the digital beam former is created and controlled by means of code written on a programmable device of the digital beam former, in this case a Xilinx FPGA [4].

2. PRINCIPLE OF DIGITAL DATA PROCESSOR

As all the operations are performed digitally, the received RF analog signal at each antenna element is first converted to digital form using high speed multi-byte parallel ADCs. The high speed samples from ADC are fed to digital down converter to get two down converted signals, a signal in-phase with the input and another in quadrature phase with the input. These I and Q components are then fed to complex multiplier where they are multiplied with the weights stored inside FPGA block RAMs. Finally the outputs of all the complex multipliers are summed to form a beam. A main lobe is produced together with nulls and side lobes. The formation of multiple beams depends highly on the sampling rate of ADCs, processor computational capacity and the operating frequency of the processor.

Thus the main components involved in implementation of digital beam forming are as follows.

- High speed parallel ADCs to convert incoming RF signal at each antenna element to the digital signal accurately.
- The digital down converters which brings down the high sample frequency from the ADCs to the baseband frequency, in order to enable to process the data at lower rates.
- Algorithm for computation of weights which are used to weigh the input signal to obtain the desired radiation pattern.
- High speed data communication path for receiving weights and sending beams for further processing and plotting.

This paper addresses several issues involved in the design and implementation of a digital beam former architecture which is developed for 16 element planar phased array radar.

3. ADAPTIVE FILTER ALGORITHMS

The important aspect of adaptive Beam forming is computation of adaptive weights on the hardware. In designing adaptive filters for radar applications, recursive least squares (RLS) and constrained recursive least squares (CRLS) [7] 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.

The remedy for the problem of ill conditioning and VLSI implementation, QRD-RLS algorithm [7] is used which is based on orthogonalization techniques such as Givens Rotation, Householder Transformation, and Gram-Schmidt Technique. QR-decompositions avoid
explicit matrix inversions and are hence more robust and well suited for hardware implementation. The QR decomposition algorithms such as Givens Rotation can be implemented in parallel and pipelined manner as required in high speed applications. The architecture used for implementation of these algorithms in parallel and pipelined manner is called as systolic array.

A systolic array is an arrangement of processors in an array where data flows synchronously between neighbors across the array, usually with different data flowing in different directions. Processors perform a sequence of operations on data that flows between them and operate concurrently.

4. OPTIMUM WEIGHT CALCULATION

The application of QR decomposition to triangularize the input data matrix results in an alternative method for the implementation of the recursive least-squares (RLS). The main advantages brought about by the recursive least-squares algorithm based on QR decomposition [6] are its possible implementation in systolic arrays and its improved numerical behavior when quantization effects are taken into account.

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

\[ R_{xx}(n)w(n) + p(n) = 0 \]  

(1)

Here, \( R_{xx} \) is the \((M-1) \times (M-1)\) 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 [6][7] of a matrix is the decomposition of the matrix into an orthogonal and a triangular matrix. This matrix decomposition can be used to solve linear systems of equations like the linear least squares problem.

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

\[ Q(n)X(n) = ((R(n))\|0) \]  

(2)

Where, \( Q(n) \) and \( R(n) \) denote \((M - 1) \times (M - 1)\) orthogonal matrix and \((M - 1) \times (M - 1)\) upper triangular matrix respectively. Since \( Q(n) \) is an orthogonal matrix the residue vector \( e(n) \) can be evaluated as:

\[ \|e(n)\| = \|((R(n)\|0)w(n) + ((u(n)))\|v(n))\| \]  

(3)

Where,

\[ ((u(n)))\|v(n)) = Q(n)d(n) \]  

(4)

The d(n) denotes the reference data of the systolic array. The RLS weight vector that minimizes \( \|e(n)\| \) can be computed by:

\[ R(n)w(n) + u(n) = 0 \]  

(5)

The advantages offered by QRD-RLS algorithm over the conventional RLS algorithm are as follows:

- Directly deals with observed data matrix
- Achieves the requirements of computational efficiency
- Achieves robust numerical stability as there is no sample matrix inversion.

5. SYSTOLIC ARRAY IMPLEMENTATION

a. Conventional QRD-RLS
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 as shown in Fig.1.

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.

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.

2) Internal Cell: Internal cell as shown in Fig.3, performs the rotation on the input data by multiplying it with the sine and the cosine from the boundary cell and generates the rotated data which is stored in its internal memory and the output which is passed to the cell in the next row.

b. INVERSE QRD-RLS
This algorithm, besides the inherited numerical robustness of its family, provide the coefficient vector at every iteration, without having to resort to the computationally tedious backward or forward substitution procedures. Conventional QRD-RLS update scheme involves two computational steps which cannot be efficiently combined on a pipelined array. Inverse QRD-RLS algorithm allows the calculation of the weight vector in completely pipelined manner.

The cells involved in the systolic array implementation of the Inverse
QRD-RLS are same as in Conventional QRD-RLS except the cells in the last row. These cells are called as Final cells or Weight Cells as the give out the weight vector.

![Inverse QRD-RLS algorithm architecture](image)

Fig. 4. Inverse QRD-RLS algorithm architecture

3) Final Cell: Final cells accept the data from the internal cells in its preceding row and the sine and cosine of the rotation angle from the boundary cell in its row. Part of the operation performed in the final cell is similar to the internal cell wherein it rotates the input data by multiplying it with the sine and cosine of the rotation angle. Along with this operation, final cell multiplies the rotated input data value with the scaling factor and then negate the result to generate the final tap coefficients.

![Final Cell](image)

Fig. 5. Final Cell

6. REALIZATION OF BEAM FORMER USING SYSTOLIC ARRAY

The Block diagram shown in Fig.6 explains the architectural features of the DBF for Four element Phased Array antenna [10].

![Block diagram of a four element DBF for multiple digital beams](image)

Fig. 6. Block diagram of a four element DBF for multiple digital beams.

This Architecture has been extended to 16 element array in this work and the same can be extended to any number of antenna arrays. The basic building blocks for this development are Digital down converters (DDC), complex adders and complex multipliers. The IF signal, generally in the range of 50MHz to 60 MHz is converted into one word digital data using 8/16 bit, 125 MS/s high speed ADCs. The digital data is received at a sampling clock of 50 MHz and then processed as follows: 16 bit high speed ADC Data is passed through a digital Mixer consisting of a 50 MHz Numerically Controlled Oscillator (NCO), a multiplier (16x16 bit), suitable low pass decimation and compensating filters (CIC and CFIR filters) of bandwidth 5 MHz to filter the entire unwanted signal outside the band and a 10 rate decimator to bring down the sampling rate to 5 MS/s for further processing. Finally the DDC output will be In phase (I) and Quadrature (Q) signals. Fig. 7 gives the architectural details.
A straightforward implementation uses two multipliers, one each for the sine and the cosine as shown in Fig. 8.

By multiplying the input data, by the quadrature sine and cosine waveforms, we achieve a frequency translation to the base band as shown in Fig. 9.

The multipliers are 16 x 16 bit signed multipliers. The lower 16-bits, of the 32-bit output, are truncated and the 16 most significant bits are used for subsequent processing. The quantization error is within 0.1% and is acceptable.

For realizing the 16 element array it is essential to have 16 different DDC modules in the complete architecture. Details are shown in the Fig 10. The complex multipliers and complex adders are implemented in hardware using VHDL [11]. To perform this complex multiplication in FPGA we need to perform equivalent floating point arithmetic operations in fixed point as the error is within limits and this is faster. The weights are calculated and stored in the memory of the FPGA. Depending upon the signal available from any direction within the range from -450 to +450, suitable weights will be applied and the required number of beams will be calculated. During the formation of the beams it is assumed that direction of arrival is known apriori as the transmit beam is scheduled by the radar computer. With respect to the direction of arrival, multiple beams are formed. The offset is fixed by the weights which are calculated and stored in the memory. With the developed architecture the weights are calculated for +/-10, +/-20 and so on.

It is required to compute the complex multiplication for several numbers of weights which will decide where the beam needs to be formed. For sixteen elements to form one beam we need to have sixteen weights and for N number of beams, N different sets of sixteen weights are required. We consider the weights are fixed and calculated offline.

The data flow architecture of the complex addition and complex multiplication are shown in Fig. 10 which is simulated using VHDL modeling and implemented on the prototype development hardware shown in Fig. 11. Summation of all the partial beams in the same digital domain, gives
the full beam $B(t)$, given by equation 6 for an $N$-element Array.

$$B(t)=k=0\sum NS_k(t) \ast W_k$$ \hspace{1cm} (6)

Where, $N$ is Number of T/R Elements, $W_k$ is Complex Weight of $K$th Element, $S(t)$ is Received Signal

The Development Hardware used to implement digital beam former architecture for 6/9 beams is shown in Fig.11. The important features of the hardware are below:

- FPGA- Virtex - 6 LX240T – 1 FF1156C [6].
- Clock
  - Onboard Oscillators: 32 MHz
  - 156.25 MHz clock oscillator for SFP
- Memory:
  - 1GB DDR2 SDRAM using MT8HTF12864H-667B.
  - 256Mb flash Memory – JS28F256P30T95 from Numonyx.
- Rocket IO interface @ 3.125 Gb/s: Four SFP connectors are provided for SFP modules.
- PCI Express : 8x lane @ 2.5 Gb/s
- USB  2.0 High Speed: Using Cypress chip CY7C68013A.

The tools used for the implementation of 16 element antenna array Digital Beam Former are given below.

- Matlab for modeling purpose.
- ISE for implementation.
- Symplify Pro for synthesis.
- ISE Simulator for functional simulation.

7. RESULTS

a. Simulation and Implementation results of fixed weights beam former

The implementation of DBF for more than 4 beams is done on Vertex 6 series FPGA[4] due to the requirement of more resources for parallel processing. On this board the input is given from DDS inside the FPGA which corresponds to 60 MHz output of the ADC. The resource utilization for this implementation is given in table 2.

<table>
<thead>
<tr>
<th>Sr. No</th>
<th>Slice Logic Utilization</th>
<th>Used</th>
<th>Available</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Number of Slice LUTs</td>
<td>86,98</td>
<td>150,720</td>
<td>57%</td>
</tr>
<tr>
<td>2</td>
<td>Number of Block RAM/FIFO</td>
<td>50</td>
<td>416</td>
<td>12%</td>
</tr>
<tr>
<td>3</td>
<td>Number of BUFG</td>
<td>6</td>
<td>32</td>
<td>18%</td>
</tr>
<tr>
<td>4</td>
<td>Number of DSP48E1</td>
<td>634</td>
<td>768</td>
<td>82%</td>
</tr>
<tr>
<td>5</td>
<td>Number of PLL ADVs</td>
<td>1</td>
<td>12</td>
<td>8%</td>
</tr>
</tbody>
</table>

1) Following are the chip scope outputs for fixed beam forming for 16 element planar antenna array (4 by 4 matrix).
• Transmission angle :-
  Azimuth=0 deg
  Elevation=20 deg
• Number of beams formed= 04
• Azimuth angle is variable
• Angle by which beams are separated=10\degree
• The beams are formed for azimuth angles=-20\degree, -10\degree, 0\degree, 10\degree.

2) Following are the chip scope outputs for fixed beam forming for 16 element planar antenna array.
• Transmission angle :-
  Azimuth=40\degree
  Elevation=10\degree
• Number of beams formed= 04
• Elevation angle is variable
• Angle by which beams are separated=10\degree
• The beams are formed for elevation angles=-10\degree, 0\degree, 10\degree, 20\degree.

The beams are plotted in the Matlab. Weights are taken from FPGA to Matlab, multiplied with scanning vector and then plotted.

3) Following are the chip scope outputs for fixed beam forming for 16 element planar antenna array (4 by 4 matrix).
• Transmission angle :-
  Azimuth=0\degree
  Elevation=30\degree
• Number of beams formed= 09
4) Following are the chip scope outputs for fixed beam forming for 16 element planar antenna array.
   - Transmission angle :
     Azimuth=30°
     Elevation=0°
   - Number of beams formed= 09
   - Elevation angle is variable
   - Angle by which beams are separated=10°
   - The beams are formed for elevation angles=-40°, -30°, -20°, -10°, 0°, 10°, 20°, 30°, 40°
B. Simulation and Implementation results of adaptive weights beam former

1) 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 = -300
Input elevation angle of arrival for desired signal = 400

Fig. 24. Beam plot for the 16 element planar antenna array using weights generated by inverse QRD-RLS in Matlab

2) 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 = 400
Input elevation angle of arrival for desired signal = 200

Fig. 25. Beam plot for the 16 element planar antenna array using weights generated by conventional QRD-RLS in Matlab.

8. CONCLUSION AND FUTURE SCOPE

We have developed a 16-element phased array multiple DBF system. The weights are calculated using the highly efficient QRD-RLS algorithm. The Virtex-VI FPGA is used for 6/9 beams, and it has enabled a remarkable reduction in the area utilization compared to the discrete and analog versions. This pipelined architecture generates multiple beams up to maximum of 9 beams simultaneously from a given array matrix of 16 elements. Conventional methods of implementation of beam forming make the system cumbersome and sensitive to temperature and other unavoidable environmental conditions. FPGA based implementation finds huge applications in modern radars as this implementation makes the system immune to the limitations that the analog methods face. At the same time, the proposed beam-forming system enjoys advantages of a reconfigurable design and low cost. The next step for enhancing the DBF system is including online weight calculation algorithm such as QRD-RLS, inside the FPGA, so as to enable radar to track the changes in continuously varying environment. Thus making the DBF system robust and efficient.

The research work is further extended to Adaptive Beam former. The adaptive weights have been calculated using has been implemented Conventional and Inverse QRD-RLS algorithm on Virtex 6VLX240T FPGA for 4 element linear phased array radar. All the arithmetic operations are carried out in single precision floating point format. Floating point representation has
provided high precision and the accurate beam formation. This QRD-RLS algorithm for 4 element linear array can be used as the component to implement the filter for large phased array antennas. High throughput rate of 1 MHz enables the radar to track the target moving with the high speed. Depending on the availability of the resources and the input sample rate; conventional or the inverse QRD-RLS algorithm can be selected. The same work can be extended for a planar array of larger array dimensions.

9. ACKNOWLEDGEMENT

We are thankful to Director LRDE for giving the opportunity of providing the hardware for testing and performance evaluation. We are also thankful to Mr. Rakesh Mehta, Director and his team from M/s BMIT, Pune for helping us to realize the proposed system hardware. We are thankful to Mr. Haseen Basha, Scientist, LRDE for helping us in the implementation of the DDC as part of the digital receiver and Digital Beam Former.

10. REFERENCES

4. Virtex- VI FPGA Data Sheets and Application notes, Xilinx Inc.
5. U. Nickel, “Fundamentals of signal processing for Phased Array Radar”