FPGA Techniques for Calibration and Beam Forming In Smart Antenna Applications

Ramanna S Havinal and Girish V Attimarad

***Abstract*—Field-programmable gate arrays (FPGA) are drawing ever increasing interest from designers of embedded wireless communications systems because of performance, power consumption and configurability. They outpace digital signal processors (DSPs), through hardware execution of a wide range of parallelizable algorithms, FPGA operational flexibility and algorithm parallelism that can adapt to variations in wireless channel statistics are utilized in smart antenna array embedded systems. FPGAs are especially suited for embedded systems because, beside an area of reconfigurable logical elements, they can also incorporate large amounts of memory, high-speed DSP blocks, clock management circuitry, high-speed input/output (I/O).We motivate the use of RLS-based techniques for adaptive antenna signal processing., where systems are required to operate reliably in the presence of strong interference and under time-varying propagation condition[5]. The digital phased array receiver presented in this paper consists of an eight-channel system with back-end FPGAs for calibration and digital beamforming processing. This paper focuses on two main topics, calibration methodologies and FPGA implementation for calibration and digital beamforming,**

***Index Terms*—Adaptive Beamforming, Calibration, (FPGAs).Calibration**

I. INTRODUCTION

THE optimal smart antenna receiver requires a front-end antenna array followed by an analog-to-digital converter (ADC) behind every element. All channels would then be processed using some type of real-time engine. Flexibility, complexity, and form factor dictate that a combination of IF sampling techniques (to minimize the amount of RF electronics) and (FPGA) technology (for size and flexibility) be used to comprise an array-based antenna receiver architecture. Such an architecture would be capable of serving a wide variety of applications from direction finding in wide-band radar processing to co-channel interference mitigation in narrow-band personal communications systems (PCS).

In generally implementation at the system level, There are a number of such issues like mutual coupling and calibration [4][7][8] must be addressed before the optimal smart antenna receiver can be practically realized.

Field-Programmable Gated Arrays (FPGA) are reconfigurable. Although FPGAs tend to be slower and to consume more power than ASIC [7], FPGA reconfigurability can benefit platform longevity by allowing design changes/upgrades even in systems already in operation. This flexibility can be effectively exploited for rapid prototyping of advanced algorithms for communications signal processing. An FPGA can, for example, implement MRC branches either sequentially, or in parallel, or anywhere in between, depending on required speed, available chip resources, and power constraints.

The QRD-RLS algorithm is chosen for adaptive control of the array due to its robust numerical properties and because it lends itself well for parallel processing as required in high speed applications, e.g., mobile broadband systems (MBS). It is demonstrated how to map this algorithm onto a highly efficient systolic array architecture entirely consisting of identical processing elements which rely on CORDIC arithmetic to carry out all the required computations, thus enabling very economic VLSI implementations. Additionally, a fast method for joint reference signal synchronisation and array adaptation which allows to control multiple independent beams in parallel. Both of these fit very nicely with the proposed hard- ware concepts and can be exploited advantageously for multipath diversity combining or for the concurrent reception of several spatially distributed users sharing the same frequency and/or time slot. Results of bit-true computer simulations are included to illustrate the performance of the proposed techniques.[6] In this article, we explain how to implement Adaptive digital Beam forming with standard FPGA and to investigate FPGA suitability for efficient smart antenna [1][6]

II. SMART ANTENNA SYSTEM OVERVIEW

Smart Antennas: A linearly arranged and equally spaced array of antennas forms the basic structure of a beam former. In order to form a beam, each user's information signal is multiplied by a set of complex weights (where the number of weights equals the number of antennas) and then transmitted from the array. The signals emitted from different antennas in the array differ in phase (which is determined by the distance between antenna elements) as well as amplitude (determined by the weight associated with that antenna).The adaptive process permits narrower beams and reduced output in other directions, significantly improving the signal-to-interference-plus-noise ratio (SINR).

Mr. Ramanna S Havinal is Assistant Professor, M.B.E Society's College of Engg., Ambajogai. (email rshavinal@rediffmail.com)
Dr. Girish V Attimarad is Assistant Professor, S D M College of Engg. and Tech, Dharwad (e-mail:shreyagiri@rediffmail.com)

A. *FPGA Based Adaptive Beamforming*

A combination of FPGAs, digital signal processing IP, and embedded processors that implement beam-forming applications The high-performance digital signal processing (DSP) blocks, embedded Nios II® processors, and logic elements (LEs) of Altera's Stratix® II FPGAs make them ideal for adaptive beamforming applications. This section describes the Altera® implementation of a Rake-Beamformer (also known as two-dimensional Rake) structure that performs joint space-time processing. As illustrated in Figure 1, the signal from each receive antenna is first down-converted to baseband, processed by the matched filter-multipath estimator, and accordingly assigned to different Rake fingers.

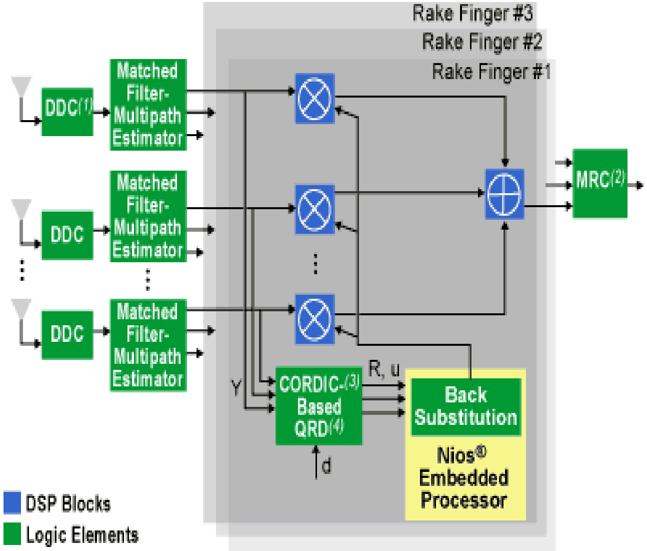

Figure 1. Adaptive Beamforming with FPGA

1.DDC: digital down converter 2.MRC: maximal ratio combining 3.CORDIC: coordinate rotation digital computer 4.QRD: QR decomposition

The beamforming unit on each Rake finger then calculates the corresponding beamformer weights and channel estimate using the pilot symbols that have been transmitted through the dedicated physical control channel (DPCCH). The QRD-based recursive least squares (RLS) algorithm is selected as the weight update algorithm for its fast convergence and good numerical properties. The updated beamformer weights are then used for multiplication with the data that has been transmitted through the dedicated physical data channel (DPDCH). Maximal ratio combining (MRC) of the signals from all fingers is then performed to yield the final soft estimate of the DPDCH data.[3]

III. CALIBRATION TECHNIQUES

One of the main advantages of having a smart antenna with digital control over every element in the array is the ease with which calibration can be implemented. A digital smart antenna has the ability to balance the channels using digital processing techniques, allowing for cheaper hardware components to be used. For the digital smart antenna system presented here, two calibration methods will be compared[2]. The first calibration method is shown in Fig. 1. A calibration signal is generated and split eight ways and passed through each channel. These eight signals are injected into each channel through a switch which is not shown in Fig.2.

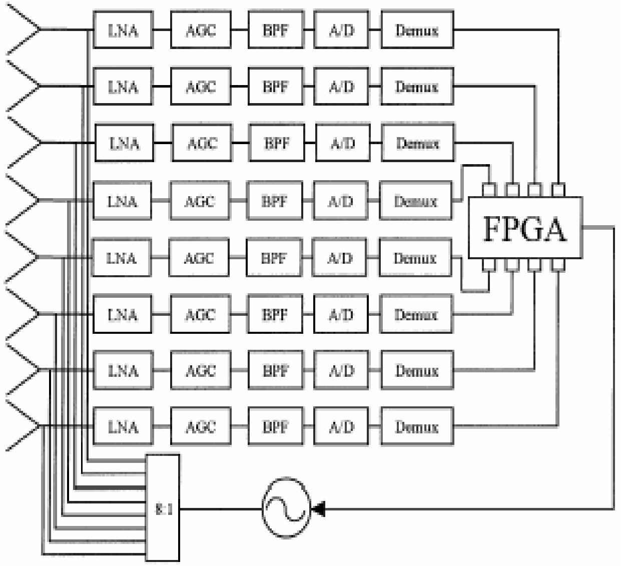

Fig. 2. Smart antenna system with calibration including only the hardware chain.

Each signal is then passed through the RF hardware, digitized, and fed to the FPGA. In the FPGA, the amplitude and phase shifts associated with each channel are computed. Given a reference channel, the appropriate phase shifts and normalized amplitudes between all channels are computed. Once these values are known, they are applied to each respective channel during the digital beamforming processing. This calibration procedure can be applied anytime the system is not collecting data. Due to temperature changes, the smart antenna system may need to calibrate several times throughout the day. The calibration signal used here is a continuous wave (CW) tone, but is not limited to such a signal and more sophisticated signals could be used [2][7].

The previous calibration method is well proven and works, but neglects the effects of the radiating antenna elements. One other concern with the above calibration method is the degradation in noise figure due to the switch in front of the LNA. To avoid both of these issues, the smart antenna system could be calibrated using the method shown in Fig.3. Here, instead of taking the calibration signal and splitting it into each channel, a small antenna probe is placed in front of the antenna in the far field at broadside. With this technique, the mutual coupling effects due to the antenna are included in the calibrated signal. The biggest concern with this method is accurately aligning the calibration antenna probe in front of the smart antenna system and also having it in the far field. Any calibration signal type could be used for this method as well.

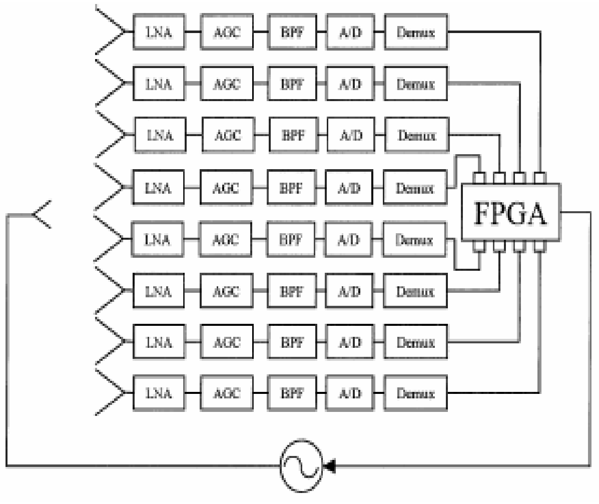

Fig.3. Smart antenna system with calibration including the antenna array and hardware chain.

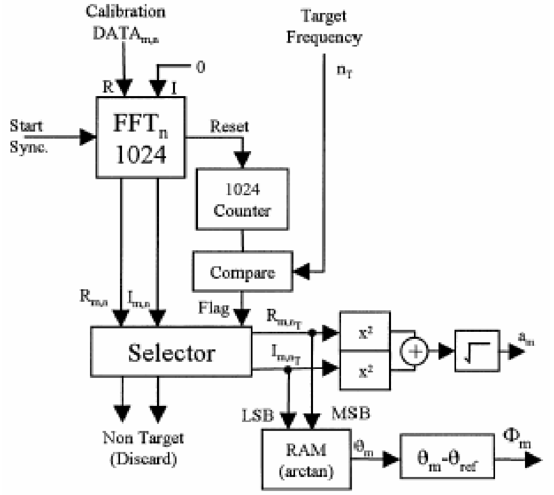

Fig. 4. FPGA procedure for computing calibration weights, amplitude, and phase, for each channel of the smart antenna system

IV. FPGA IMPLEMENTATION

The FPGA implementation of the smart antenna system for performing calibration and digital beamforming for all eight channels,The FPGA platform (Wildstar VME, Annapolis Microsystems) used consists of five Xilinx Virtex-E (XCV1000E)[2][7] FPGAs, where each FPGA has 1.5 million gates, 12 288 slices, and 393 216 b of block RAM can be used.The FPGAs were programmed using a combination of Xilinx cores and custom very high-speed integrated-circuit (VSIC) hardware design language (HDL) (VHDL) code. The limitations in implementing digital beamforming processing on the FPGA platform include the I/Os between the individual FPGA chips and after the output, and the device size and speed needed to implement the algorithms[1][2]

A. Calibration

The FPGA calibration procedure for a single channel is shown in Fig. 4. Given 8-b time-domain data samples from the ADC, the FFT performs a Cooley–Tukey radix-4 decimation- in-frequency FFT (comprised of five ranks of 256 butterfly operations) and outputs 16-b frequency domain data samples for both the real and imaginary components (imaginary inputs are tied to zero).

These outputs are fed into a selector that picks out the complex pair corresponding to a target frequency (occupied by the calibration tone). The magnitude and phase am and Φm of this pair are then computed and stored as a calibration amplitude-phase pair. Note that Φm is computed from the phase of the individual channel θm with respect to the reference channel θref. The square root operation is implemented using the coordinate rotation digital computer (CORDIC) algorithm .It would be a straightforward process to merely search for the frequency containing the most signal energy and perform the calibration there. This would allow a swept frequency signal to be delivered to the front end of each channel and provide for rapid wide-band calibration

B. Digital Beam forming

The digital beamforming FPGA processing is done in two parts. First, the calibration weights are applied using and second, the eight channels are beam formed through a matrix multiplication.The calibration correction implementation is shown in Fig. 5. Given a current calibration set (comprised of eight amplitude weights and eight phase shifts), it is a straightforward though nontrivial process to apply the calibration to data arriving from the ADCs. Each calibration block is designed to take inputs from four ADCs (for eight array elements, two calibration blocks are needed). Each calibration block consists of four RAM blocks which allow for four channels of simultaneous data to be captured in near real time. Each calibration block consists of a 1024-point FFT (shared between the four RAM blocks using a multiplexer), and a sin/cos lookup table. After the 8-b ADC data is captured in RAM, each channel is individually processed through the FFT and 16-b frequency-domain data is generated

These outputs (real and imaginary) are each multiplied by am the for the current channel, thus correcting for any deviations in magnitude. Concurrently, Φm (constant phase shift per channel stored as a 10-b number) is sent to a sin/cos lookup table with the appropriate sign to account for positive and negative frequencies. The lookup table outputs 12-b values which alter the real and imaginary amplitude corrected signals by performing a complex multiplication (requires four real multiplies and two additions). Output samples are now compensated for unwanted amplitude and phase variations using the weights computed during the calibration process. Each channel (four per block) goes through the calibration process serially and is stored in RAM as complex data until all channels are processed. All of this processing (four channels) is located on a single FPGA using 25% of the slices, 33% of the block RAM. The calibration for eight antenna elements is performed on two FPGAs and subsequently passed to a third FPGA for the beam forming operation. The data is passed between the FPGAs in parallel mode because of the requirement to have all channel data present simultaneously for the beamforming operation.

The data is passed between the FPGAs in parallel mode because of the requirement to have all channel data present simultaneously for the beamforming operation. The beamforming operation requires a large complex matrix multiplication given by

$[T]_{25 \times 1024} = [A]_{25x8} [Q^{CAL}]_{8x1024}$

where Q^{CAL}represents the calibrated FFT data of all eight channels, A represents the beam scanning matrix weights and T represents the beamformed data output. Note that there are a total of 1024 samples per data stream, 25 scan angles (60° to 60° in 5 steps), and eight channels. Eight new complex samples are received every clock period from the data streaming out of the channel correction blocks.

These eight complex numbers comprise one column of the Q^{CAL} matrix. The clock period when they arrive at the beamforming block indexes the columns of the Q^{CAL} matrix (frequency).

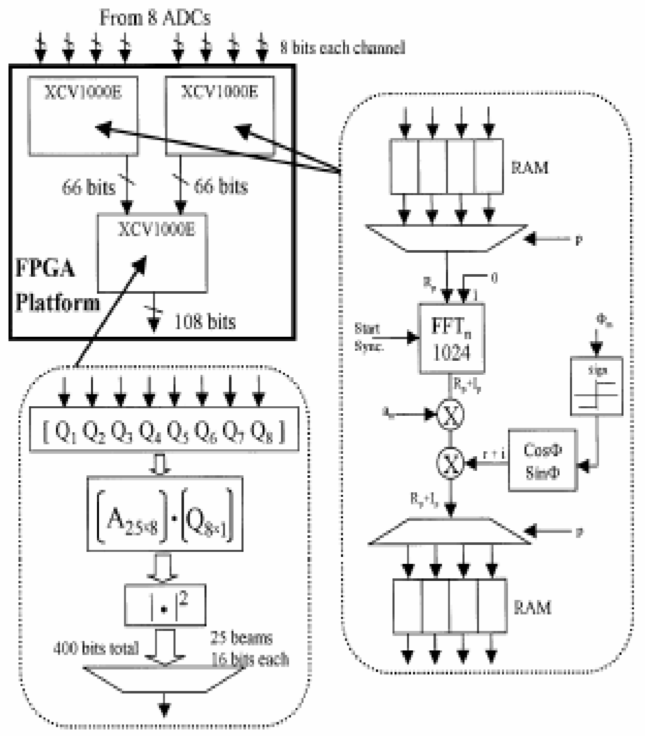

Fig. 5. FPGA architecture for applying channel calibration weights and computing digital beam forming matrix.

Thus, each clock period needs only to calculate the results for one column of the T matrix by using only one column from the Q^{CAL} matrix. Unfortunately, this calculation requires each value in the matrix and therefore the entire matrix must be stored within the FPGA. The Q^{CAL} matrix data and matrix data are both complex and thus complex multiplies and twice the storage are required. For direction finding, only the magnitude of the power spectrum is necessary and therefore the real and imaginary parts of the output are squared and summed.

V. ADAPTIVE ALGORITHMS:

Adaptive signal processing algorithms such as least mean squares (LMS), normalized LMS (NLMS), and recursive least squares (RLS) have been used in a number of wireless applications such as equalization, beam forming and adaptive filtering. These all involve solving for an over-specified set of equations, Among the different algorithms, the recursive least squares algorithm is generally preferred for its fast convergence. The least squares approach attempts to find the set of coefficients that minimizes the sum of squares of the errors, in other words:

$$\{ \sum_{m} e(m)^2 \} \qquad 1$$

Representing the above set of equations in the matrix form, we have: $X\mathbf{c} = \mathbf{y} + \mathbf{e}$ (1) where X is a matrix (mxN, with $m>N$) of noisy observations, **y** is a known training sequence, and **c** is the coefficient vector to be computed such that the error vector **e** is minimized.[4][5]

QR-decompositions shown in fig 6, avoid explicit matrix inversions and are hence more robust and well suited for hardware implementation. FPGAs are the preferred hardware for such applications because of their ability to deliver enormous signal-processing bandwidth. FPGAs provide the right implementation platform for such computationally demanding applications with their inherent parallel-processing benefits (as opposed to serial processing in DSPs) along with the presence of embedded multipliers that provide throughputs that are an order of magnitude greater than the current generation of DSPs. The presence of embedded soft processor cores within FPGAs gives designers the flexibility and portability of high-level software design while maintaining the performance benefits of parallel hardware operations in FPGAs. [2][4][8]

A. QRD Algorithms:

The aim of the **RLS** algorithm is to minimize the sum of exponentially weighted squared errorsThe choice for fast and numerically robust RLS filtering is based upon orthogonal triangularisation of the input data matrix via *QR* decomposition (QRD) [5]. In situations where the data is to be processed on a sample by sample basis (as opposed to block-wise) QR-updating through a sequence of Givens rotations is usually employed. The resulting QRD-RLS algorithm requires $O(M^2)$ operations per updating step. Its numerical robustness stems from the fact that it operates directly on the incoming data matrix, circumventing the need to form an estimate of the correlation matrix *Rxx[t]* whereby the condition number of the data matrix is squared and the required word length doubled. we briefly review the QRD-RLS algorithm is used to compute the *a posteriori (a priori)* estimation error *e[t](ε[t])* without explicit computation of the weight vector. The least squares algorithm attempts to solve for the coefficient vector **c** from X and y. To realize this, the QR-decomposition algorithm is first used to transform the matrix X into an upper triangular matrix R (N x N matrix) and the vector **y** into another vector **u** such that

*R***c**=**u**. The coefficients vector **c** is then computed using a procedure called *back substitution*, which involves solving these equations:[1][4][7]

$$C_N = U_N / R_{NN} \qquad 2$$

$$c_i = \frac{1}{R_{ii}}\left(u_i - \sum_{j=i+1}^{N} R_{ij} C_j\right) \text{for } i = N-1, \ .1 \qquad 3$$

The QRD-RLS algorithm flow is depicted in Figure 5.

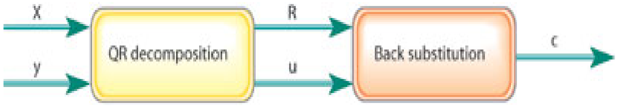

Figure 6: QR-decomposition-based least squares

B. *CORDIC Based Decomposition*

The QRD-RLS weights update algorithm involves decomposing the input signal matrix **Y** into **QR**, where **Q** is a unitary matrix and **R** is an upper triangular matrix. This is achieved using a triangular systolic array of CORDIC blocks, as shown in Figure 6. Each CORDIC block operates in either vectoring or rotating modes and performs a series of micro rotations through simple shift and add/subtract operations and can run at speeds of 300 MHz.[4][7][8].The **R** matrix and **u** vector (transformed reference signal vector **d**) are recursively updated for every new row of inputs entering the triangular array. The triangular systolic array can be further mapped into a linear array with reduced number of time-shared CORDIC blocks—as illustrated in Figure 7—providing a trade-off between resource consumption and throughput. .[3][7][8]

The QR-decomposition of the input matrix X can be performed, as illustrated in Figure 7, using the well-known systolic array architecture. The rows of matrix X are fed as inputs to the array from the top along with the corresponding element of the vector y. The R and u values held in each of the cells once all the inputs have been passed through the matrix are the outputs from QR-decomposition. These values are subsequently used to derive the coefficients using back substitution technique.[1][8]

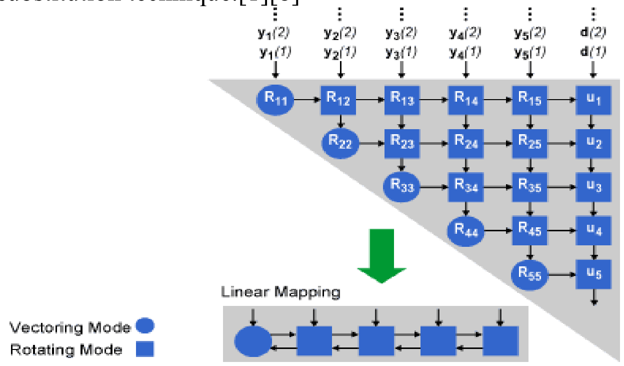

Figure 7.Triangular Systolic Array for CORDIC-Based QRD-RLS

Each of the cells in the array can be implemented as a coordinate rotation digital computer (CORDIC) block. CORDIC describes a method of performing a number of functions, including trigonometric, hyperbolic, and logarithmic functions.[2] The algorithm is iterative and uses only add, subtract, and shift operations, making it attractive for hardware implementations.[3][5][8]

For complex inputs, only one CORDIC block is required per cell. Many applications involve complex inputs and outputs to the algorithm, for which three CORDIC blocks are required per cell. In such cases, a single CORDIC block can be efficiently timeshared to perform the complex operations. Direct mapping of the CORDIC blocks onto the systolic array, as shown in Figure 6, consumes a substantial amount of an FPGA's logic but yields enormous throughput that's probably overkill for many applications. The resources required to implement the array can be reduced by trading throughput for resource consumption via mixed and discrete mapping schemes.[3][7][8]

C. *Weights and Measures*

The back-substitution procedure operates on the outputs of the QR-decomposition, involves mostly multiply and divide operations that can be efficiently executed in FPGAs with embedded soft processors[6][7]Some FPGA-resident processors can be configured with a 16x16 -> 32-bit integer hardware multipliers. The software can then complete the multiply operation in a single clock cycle. Since hardware dividers generally are not available, the divide operation can be implemented as custom logic block that may or may not become part of the FPGA-resident microprocessor. Between the multiply and divide accelerators, back-substitution becomes easy and efficient [1][2]8]

The final beamformer weights vector **w** is related to the **R** and **u** outputs of the triangular array as **Rw=u**. Because **R** is an upper triangular matrix, **w** can be solved using a procedure called back substitution that can be implemented in software on the flexible embedded Nios processor

VI. CONCLUSION

This paper has shown that a smart antenna system with digital beamforming and FPGA processing is possible In this paper we have described systolic array architectures for high speed, high performance RLS-based adaptive beamforming. The QRD-RLS algorithm has been identified as a very promising candidate for adaptive weight control . CORDIC arithmetic is advocated for hardware implementation of the individual cells in a possible processor array realisation, since the same number of cycles are required to perform all the needed operations resulting in a truly systolic data flow Two calibration methods were compared and both were demonstrated with measured data captured by the smart antenna receiver system. With advances in FPGAs and ADCs, it will soon be practical and cost-effective to build larger smart antenna systems with reconfigurable applications.

VII. . REFERENCES

Technical Reports:

[1] Beamforming smart antenna using FPGA's By Deepak Bopanna and Asif Batada

[2] Performance Characterization of FPGA Techniques for Calibration and Beamforming in Smart Antenna Applications Todd W. Nuteson, Jeffrey E. Stocker, James S. Clark, IV, Dewan S. Haque, and Gregory S. Mitchell, , IEEE Transactions On Microwave Theory And Techniques,Vol.50, No.12, December 2002 3043

[3] Nuteson Et Al.: Performance Characterization Of FPGA Techniques In Smart Antenna Applications

Periodicals:

[4] Zhong Mingqian, Tim, A.S.Madhukumar, and Francois Chin, "QRD-RLS adaptive equalizer and its CORDIC-based implementation for CDMA systems," International Journal on Wireless & Optical Communications, Vol.1, No.1 (2003) 25-39.

[5] Volder, J. "The CORDIC trigonometric computing technique," IRE Trans. Electron. Comput., Vol. EC-8, pp. 330-334, 1959.

[6] Rader,C.M."VLSI systolic arrays for adaptive nulling," IEEE Sig.Proc.Mag, Vol.13, No.4, pp.29-49, 1996.

[7] J. Larocque, J. Litva, and J. Reilly, "Calibration of a smart antenna for carrying out vector channel sounding at 1.9 GHz," in Proc. 8th Annu. Virginia Wireless Personal Commun. Tech. Symp., June 1998, pp. 293–302.

[8] T.W. Nuteson, J. S. Clark, IV, D. S. Haque, and G. S. Mitchell, "Digital beamforming and calibration for smart antennas using real-time FPGA processing," in IEEE MTT-S Int. Microwave Symp. Dig., vol. 1, June 2002, pp. 307–310.

[9] K. R. Dandekar, H. Ling, and G. Xu, "Effect of mutual coupling on direction finding in smart antenna applications," IEEE Electron. Lett., vol. 36, pp. 1889–1891, Oct. 2000.

VIII. BIOGRAPHY:

Ramanna S Havinal received the B.E(ECE), M.E(Digital Electronics) degree from Karnataka university Dharwad and Pursuing his Ph.D degree in Microwave Communication. He is currently working as Ass.Prof at M.B.E.Society's College of Eng, Ambajogai. His research interests include Digital Signal Processing, Wireless Comminication, VLSI

Dr Girish V Attimarad received the B.E(ECE), M.E(Digital electronics) degree from Karnataka university Dharwad and Obtained his Ph.D degree from Delhi College of Engineering, Delhi. He is currently working as Ass Prof at SDM Society's College of Engg and Tech, Dharwad His research areas are Digital Signal Processing, Wireless Communication, VLSI, Electromagnetics