Implementation of Smart Antenna Adaptive Beamforming Algorithm In FPGA

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), as well as support for external memory and high speed networking and communications bus standards. Power consumed in embedded systems is very less The objective of this paper is to investigate FPGA suitability for efficient smart antenna array system. 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 objective of the smart antenna beamformer is to maximize the gain of the antenna array in the direction of arrival of the desired signal and minimize it in the direction of interferences by means of the beam pattern control**

Index Terms: **Adaptive, antenna, eigen-combining, embedded, FPGA, receiver**

I. INTRODUCTION

OVER the last few years the demand for wireless communications has grown beyond all expectations. Adaptive arrays, or “smart” or “intelligent antennas”, are currently attracting great attention, since they increased spectral efficiency as well as improved performance and quality of service in future generation high capacity mobile networks and high data rate radio LANs.

Smart antenna uses a variety of new signal-processing algorithms, the system takes advantage of its ability to effectively locate and track various types of signals to dynamically minimize interference and maximize intended signal reception. Smart antennas is used to locate and track stationary or mobile users of interest while suppressing other user and any other co-channel interference

Field-Programmable Gated Arrays (FPGAs) are reconfigurable. Although FPGAs tend to be slower and to consume more power than ASICs[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[3].

Smart-antenna technology requires a lot of processing bandwidth, in the neighborhood of several billion multiply-and-accumulate (MAC) operations per second. Such computationally demanding applications can quickly exhaust the processing capabilities of many DSPs. Some FPGA chips with embedded DSP blocks, on the other hand, provide throughput in excess of 50 GMAC/sec, offering a high-performance alternative for beam-forming applications.FPGAs with embedded processors are flexible by nature, providing options for various adaptive signal-processing algorithms[1][3].

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 allows to control multiple independent beams in parallel are discussed.

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.[5] In this article, We explain how to

Mr. Ramanna S Havinal is Assistant Professor, M.B.E Society’s College of Engg., Ambajogai (e-mail: rshavinal@rediffmail.com)

Dr. Girish V Attimarad is Assistant Professor, S D M College of Engig and Tech, Dharwad (e-mail: shreyagiri@rediffmail.com,)

implement Adaptive digital Beam forming with standard FPGA and to investigate FPGA suitability for efficient smart antenna [1][3][5]

II. SMART ANTENNAS ADAPTIVE EAMFORMING

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).A smart-antenna system, as shown in Figure 2, includes an array of antennas that together direct different transmission/reception beams toward each cellular user in the system

A smart antennas system, is shown in Figure1.In beamforming, each user's signal is multiplied with complex weights that adjust the magnitude and phase of the signal to and from each antenna. This causes the output from the array of antennas to form a transmit/receive beam in the desired direction and minimizes the output in other directions. Through adaptive beamforming, antenna can form narrower beams towards the desired user and nulls towards interferer, considerably improving the signal-to-interference-plus-noise ratio.

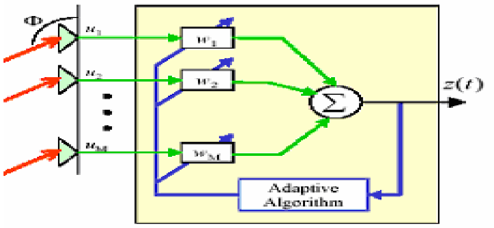

Figure 1. Smart Antennas System

III. 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 2 , 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.

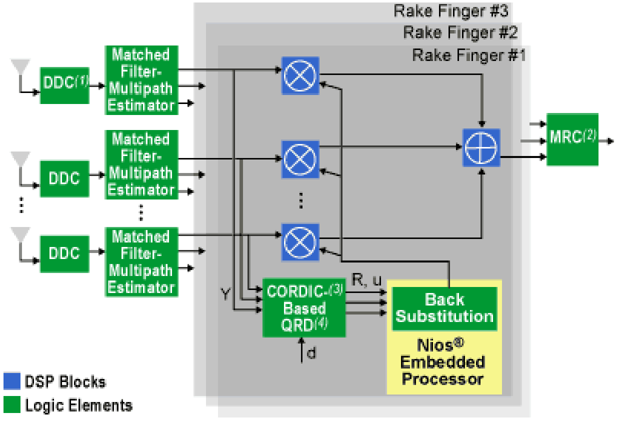

Figure 2. Adaptive Beamforming with FPGA

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.[1][3]

A. *Complex Weight Multiplication Using DSP Blocks*

Applying complex weights to the signals from different antennas involves complex multiplications that

map well onto the embedded DSP blocks available for many FPGAs. The Figure 3 shows DSP blocks with a number of multipliers, followed by subtractor / adder/ accumulators, with registers for pipelining. Such a structure lends itself to complex multiplication and routing required in beam-forming designs[1][3][6]

The application of complex weights to the signals from different antennas involves complex multiplications that map well onto the embedded DSP blocks available in Stratix II devices. Each DSP block can be operated at more than 370 MHz and has a number of multipliers, followed by adder/ subtractor/ accumulators, in addition to registers for pipelining. With these features, Stratix II devices can efficiently implement complex multiplications and reduce the amount of overall logic and routing required in beamforming designs.

B. Adaptive Algorithms

Adaptive signal processing algorithms such as least mean squares (LMS), normalized LMS (NLMS), and recursive least squares (RLS) have historically 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, as shown below, where m > N:

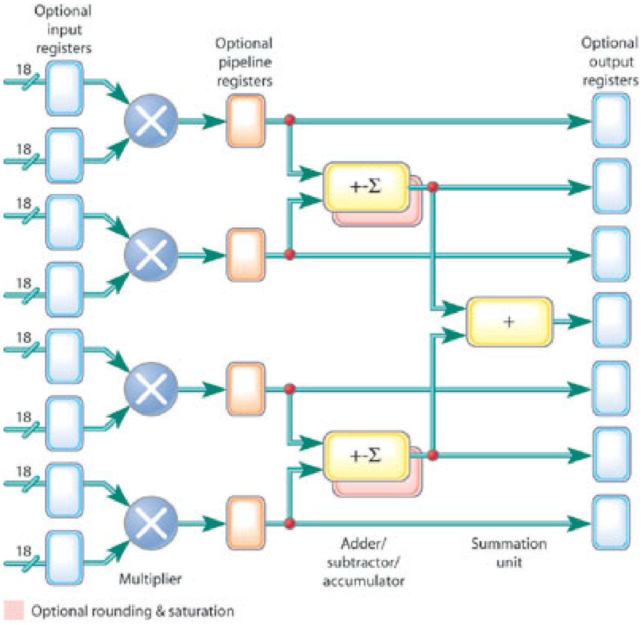
Optional input registers
Optional pipeline registers
Optional output registers
18
18
18
18
18
18
18
18
+-Σ
+
+-Σ
Multiplier
Adder/ subtractor/ accumulator
Summation unit
Optional rounding & saturation

Figure 3: DSP block architecture

$$x_1(1)c_0 + x_2(1)c_1 + \ldots\ldots x_N(1)c_N = y(1) + e(1)$$
$$x_1(2)c_0 + x_2(2)c_1 + \ldots\ldots x_N(2)c_N = y(2) + e(2)$$
$$\vdots$$
$$x_1(m)c_0 + x_2(m)c_1 + \ldots\ldots x_N(m)c_N = y(m) + e(m)$$

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.[3][8]

Direct computation of the coefficient vector **c** involves matrix inversion, which is generally undesirable for hardware implementation due to numerical instability issues. Matrix decomposition based on least squares schemes, such as Cholesky, LU, SVD, and QR-decompositions, avoid explicit matrix inversions and are hence more robust and well suited for hardware implementation. Such schemes are being increasingly considered for high-sample-rate applications such as digital predistortion, beam forming, and MIMO signal processing. 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.

QR-decompositions shown in Figure 4 avoid explicit matrix inversions and are hence more robust and well suited for hardware implementation. Such schemes are being increasingly considered for high-sample-rate applications such as digital predistortion, beam forming, and MIMO signal processing. 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][3][7]

C. QRD-RLS Algorithms

As described in Pattan's book,1 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 Rc=u. The coefficients vector c is then computed using a procedure called back substitution, which involves solving these equations:[1][3][6].The QRD-RLS algorithm flow is depicted in Figure 5.

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

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

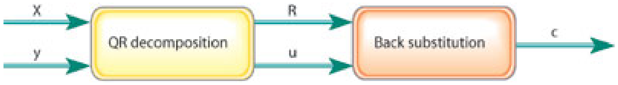

Figure 4: QR-decomposition-based least squares

D. CORDIC Based QR 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.[3][6][7]

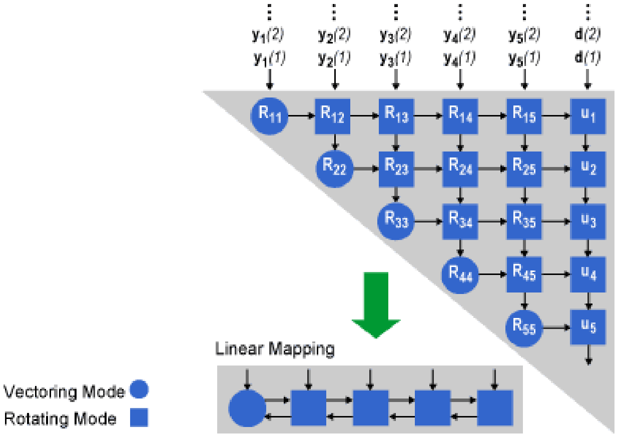

Figure 5.Triangular Systolic Array Example for CORDIC-Based QRD-RLS

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 5—providing a trade-off between resource consumption and throughput. [6][7][9]

The QR-decomposition of the input matrix X can be performed, as illustrated in Figure 6, 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.[9]

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. The number of iterations depends on the input and output precision, with more iterations being needed for more bits.[7][9]

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.[6][7][9]

In a mixed mapping scheme, the bottom rows in the systolic array are moved to the end of the top rows to make it possible to have the same number of cells in each row. Then, a single CORDIC block can perform the operations of all the cells in a row, with the total number of CORDIC blocks required being equal to the total number of rows. This is called mixed mapping because each CORDIC block has to operate in both vectorize and rotating modes[3][4]

In a discrete mapping scheme, at least two CORDIC blocks are required. One block is used purely for vectorize operations while the other is used for rotate operations.4 This single characteristic of the processor enables the realization of many gains from hardware optimization, such as enabling tradeoffs between speed and resource consumption on the FPGA[3][4],[5]

E. 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. 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

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. For an example eight antennas system, the beamformer weights for a Rake finger can be solved via back substitution in approximately 0.2 ms using the Nios operating at 100 MHz. The computation time can be lowered to 3 µs by implementing the back substitution on a hardware peripheral controlled by the Nios processor.

Moreover, the Nios processor offers a flexible platform to implement other adaptive weight update algorithms such as least mean squares (LMS) and normalized LMS[5][3].

F. FPGA Power Usage Consideration

Power loss in FPGA devices can be categorized as static and dynamic [1][3]. Static(stand-by) power is consumed by the chip when no input signals are exercised [1]. This loss occurs due to transistor leakage, which is frequency-independent, but highly dependent on junction temperature and transistor size. Dynamic power is consumed in normal operation, due to the charging and discharging of the internal capacitive loads, and is proportional to gate output load, square of the supply voltage, clock frequency, and gate switching activity [1][5].Although the supply voltage has decreased significantly in newer process technologies, high operating frequencies can still yield significant dynamic power losses [4]. A tight power budget may thus limit clock speed [1][2]. FPGA structure is designed to minimize power losses. Nonetheless, power-aware application design can also increase efficiency, e.g., by using gated clock signals and thus virtually turning off unnecessary chip sections [1][4].Gating as close as possible to the clock source is good practice since clock signal trees are important dynamic power consumers. On the other hand, static power consumption can be reduced by adaptive distribution of available FPGA resources.

IV. CONCLUSIONS

In this paper we have described algorithms and 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 due to its positive numerical characteristics and the ability to perform the required calculations in a parallel and pipelined manner. 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.[5]

Performance enhancement and reduce power consumption achieved for FPGA-based embedded smart antenna arrays FPGA flexibility and their wide range of on-chip resources can thus yield very efficient embedded system implementations of adaptive antenna which are based on parallelizable algorithms,

V. REFERENCES

Technical Reports:

[1] [1]FPGA-based Communications Receivers for Smart Antenna Array Embedded Systems Constantin Siriteanu, Steven D. Blostein, and James Millar

[2] [2]Algorithms And VLSI Architectures For RLS-Based Time Reference Beamforming In Mobile Communications Bruno Haller * Communication Technology Laboratory Swiss Federal Institute Of Technology (Eth) Ch-8092 Zurich, Switzerland

[3] [3] Beamforming smart antenna using FPGA's By Deepak Bopanna and Asif Batada

[4] [4] FPGA-based Communications Receivers for Smart Antenna Array Embedded Systems Constantin Siriteanu, Steven D. Blostein, and James Millar

Periodicals:

[5] 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 Commns, Vol.1, No.1 (2003) 25-39.

[6] [6]Volder, J. "The CORDIC trigonometric computing technique," IRE Trans. Electron. Comput., Vol. EC-8, pp.330-334,1959.

[7] [7]Ghtbody, G., R.L.Walke, and R.Woods, J.McCanny, "Novel mapping of a linear QR architecture," Proc. ICASSP,volIV,pp.1933-6,1999.

[8] [8]Walke, R.L. and R.W.M. Smith, "Architectures for adaptive weight calculation on ASIC and FPGA," 33rd Asilomar Conference on Signals, Systems and Computers,1999.

[9] [9]Rader, C.M. "VLSI systolic arrays for adaptive nulling," IEEE Sig.Proc.Mag, Vol.13, No.4, pp.29-49, 1996.

VI. 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'College of Engineering and Technology Dharwad .His research areas are Digital Signal Processing, Wireless Communication, VLSI Electromagneti