# Digital Beamforming and Calibration for Smart Antennas using Real-Time FPGA Processing

Todd W. Nuteson, James S. Clark IV, Dewan S. Haque, and Gregory S. Mitchell

The Aerospace Corporation, Chantilly, VA, 20151, USA

*Abstract*— **A real-time field programmable gate array (FPGA) implementation is presented for a smart antenna application employing digital beamforming (DBF). The DBF is performed at the RF signal carrier frequency by means of bandpass (IF) sampling, using high-speed 8-bit analog-to-digital (A/D) converters. The digital phased array receiver presented in this paper consists of an 8-channel system with back-end FPGAs for real-time calibration and DBF processing. The system operates in the L-band (1.8 GHz to 2 GHz) with various bandwidths depending on the application and FPGA processing requirements. Two main topics are discussed in this paper: 1) calibration methodologies and 2) FPGA implementation for calibration and DBF. Results presented include real measured data that was collected with the system and processed via FPGAs.**

## I. Introduction

The ultimate smart antenna would be a system with a front-end phased array antenna and an A/D behind every element, where all elements would be processed in real-time via FPGAs. A system like this would be the most flexible platform of any phased array antenna architecture, hence making it amenable for a wide variety of applications. With current technologies, it is possible to build such a system, although the cost is high. The key technology driver for this system is the FPGA, because for this system to be used in practice, it will need to process and perform all desired functions in real-time. Another alternative to FPGAs, would be building an application specific integrated circuit (ASIC) to perform the desired operation. This approach would provide a faster and more efficient solution when compared to the FPGA approach, but the down side is the loss of flexibility. For example, with an FPGA based processing approach the system could be reconfigured from a wideband radar to a narrowband PCS with co-channel interference mitigation without any hardware modifications.

After the FPGA, the next key element of this system would be the A/D. It is important to note that the ultimate smart antenna system would digitize at the RF carrier frequency instead of doing the traditional RF-IF analog down conversion. By doing so, the complexity of the analog hardware is greatly reduced simply by the elimination of most hardware. All that is required before the A/D is a low noise amplifier (LNA), automatic gain control (AGC), and a bandpass filter (BPF). An AGC is required for proper loading of the A/Ds [1]. Commercial high-speed A/Ds are available today that operate up to 1.5 Gsps with 8-bit resolutions. For most situations, this is more than adequate, but for systems that require more than 600 MHz of bandwidth, faster A/Ds would be required.

This paper will present an ultimate smart antenna as defined above, consisting of 8-channels [1], that was built in hardware and performs real-time FPGA processing. First, an overview of the system will be presented describing the hardware elements of the smart antenna. Following the system overview, calibration methodologies [2]-[8] that were used to calibrate the smart antenna will be discussed. The last section of the paper will focus on the FPGA implementation including calibration and DBF. Measured results are presented for both calibration techniques and FPGA processed data.

## II. System Overview

A block diagram of the 8-channel digital smart antenna receiver system is shown in Fig. 1. Maximum operational system bandwidth is 200 MHz, from 1.8 GHz to 2.0 GHz, but due to FPGA limitations the actual real-time processed bandwidth will be lower than this. Bandwidth limitations due to the FPGAs will be discussed in the next section. The system analog front-end consists of an 8-element linear half-wave dipole array where each element has behind it a LNA, an AGC, and a BPF. After each analog channel, the RF signal is digitized using a high-speed A/D converter with a maximum sampling frequency of 1.5 GHz. The final stage of the system is the FPGA. Before the digitized data can be sent to the FPGA, it must first be demuxed in order to get the clock rate down to speeds that the FPGA can process. As an example, for full system capability with 200 MHz bandwidth, the best sampling frequency with the most guard band would be 845 MHz [9]. This produces a sampled IF center frequency of 210 MHz. With a 845 MHz sampling clock rate a demux of 1:8 would be necessary to slow the data down to 105.625 MHz, allowing it to be implemented in a Xilinx Virtex-II FPGA [10].

## III. System 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. In a practical system, it is difficult to balance the amplitude and phase of every channel over temperature and frequency. Accurate alignment of channels would require high precision hardware components which would accrue system cost. For comparison, a digital smart antenna has the ability to balance the channels in processing, allowing for cheaper hardware components to be used. For the digital smart antenna system presented here, two calibration methods will be compared.



0-7803-7239-5/02/$10.00 © 2002 IEEE

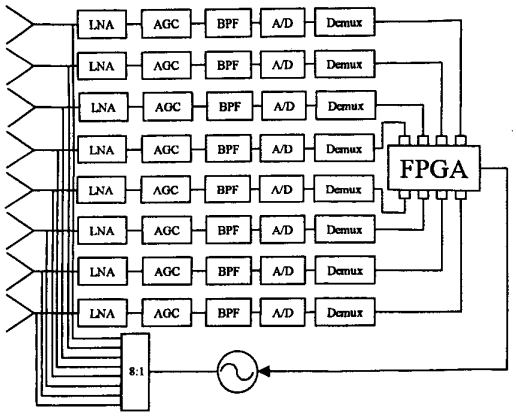


Fig. 1. Smart antenna system with calibration method 1.

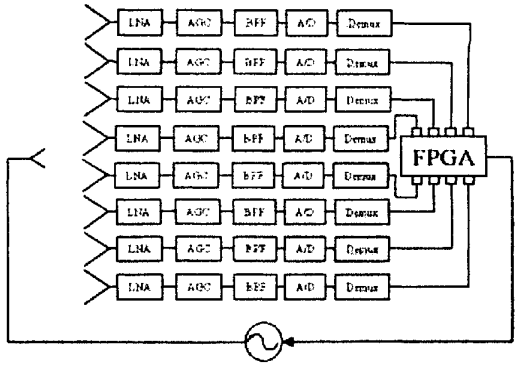


Fig. 2. Smart antenna system with calibration method 2.

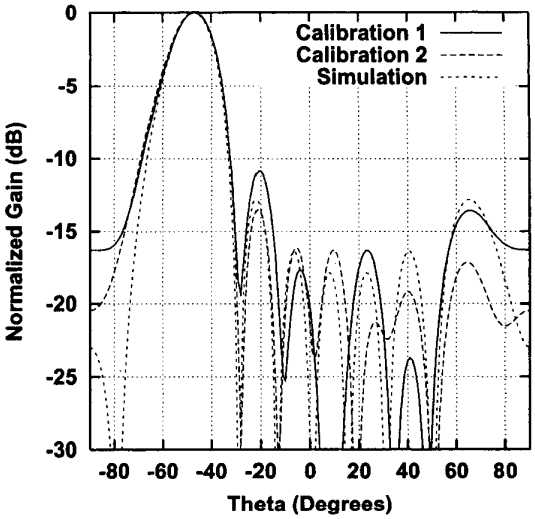


Fig. 3. Antenna pattern cut at -47 degree scan.

The first calibration method is shown in Fig. 1. Here a calibration signal is generated and split out eight ways and sent through each channel. These eight signals are injected into each channel through a 10 dB directional coupler which is not shown in Fig. 1. Each signal is then passed through each hardware component, digitized, and fed to the FPGA, where in the FPGA the amplitude and time-delays associated with each channel are computed. Given a reference channel (predetermined to be the slowest channel in the system), the appropriate time shifts between all channels are computed and the amplitudes are all normalized with respect to the largest amplitude. Once these values are known, they are applied to each respective channel during the DBF processing. This calibration procedure can be applied at anytime when the system is not collecting real data. For example, due to temperature changes throughout a day, the smart antenna system may need to calibrate several times throughout the day. The calibration signal used here was a CW tone but is not limited to such a signal. For example, more sophisticated signals could be used [3], [8].

The previous calibration method is well proven and works, but neglects the effects of the radiating antenna elements. The mutual coupling of the antenna elements could be added [11], but they would have to be known. One other concern with the above method is the hit in noise figure due to the 10 dB directional coupler in front of the LNA. To avoid both of these issues, the smart antenna system could be calibrated using the method shown in Fig. 2. 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, each channel has mutual coupling effects due to the antenna. 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. Again for this calibration method, any calibration signal could be used.

An experiment was performed to compare both of these calibration techniques shown in Figs. 1,2. Note that for these experiments, the data was processed off-line and not with the FPGAs. The calibration signal was a 1.92 GHz CW tone and the sampling rate of the A/Ds was 1 Gsps. With this calibration signal, the smart antenna was calibrated for both methods. Next a signal centered 1.92 GHz was transmitted to the smart antenna system at an angle of -47 degrees. This signal was captured and processed off-line using the calibration weights computed from each method. Fig. 3 shows the processed measured results, where it can be seen that calibration method 2 compares more favorably with theory (simulation) than calibration method 1.

## IV. FPGA Implementation

The next two subsections will present the FPGA implementation of the smart antenna system for performing calibration and DBF for all 8-channels. Xilinx Virtex-E (XCV1000E) [12] FPGAs were used, where each FPGA has 1.5 million gates, 12,288 slices, and 393,216 bits of block RAM. The FPGAs were programmed using VHDL and a

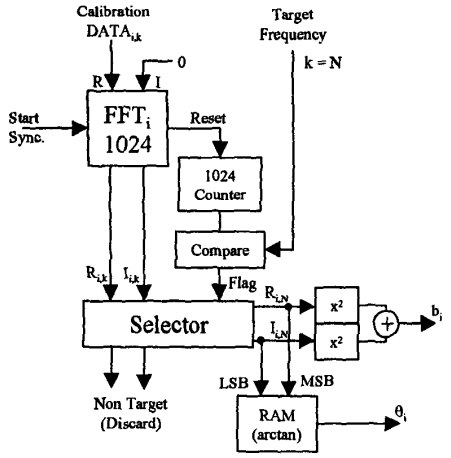


Fig. 4. FPGA procedure for computing calibration weights.

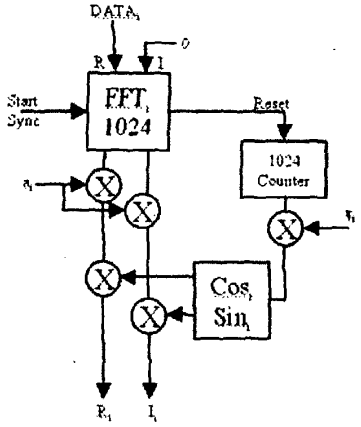


Fig. 5. FPGA procedure for applying calibration weights.

total of 5 Xilinx Virtex-E FPGAs are available in the system.

The largest resource in the FPGAs is taken up by the 1024 point FFT operation (see Figs. 4,5). The smart antenna system needs a total of 8 FFT operations, one for each channel. With a Xilinx Virtex-E FPGA, 4 FFTs can be implemented on one FPGA, so therefore 2 FPGAs are needed for all 8 channels. Each FFT uses 21% of the slices and 25% of the block RAM. Note that with a state of the art FPGA, such as the Xilinx Virtex-II (XC2V10000) [10], 30 1024 point FFTs would fit on 1 FPGA chip. The Xilinx Virtex-II has 10 million gates, 61,000 slices, and 3,538,944 bits of block RAM.

### *A. Calibration*

The FPGA calibration procedure for a single channel is shown in Fig. 4. Given 8-bit time domain data samples from the A/D, the FFT performs a Cooley-Tukey radix-4 decimation-in-frequency FFT(comprised of 5 ranks of 256 butterfly operations) and outputs 16-bit 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 calibration tone). The magnitude and phase of this pair are then computed and stored as a calibration amplitude-delay pair. There are many ways this calibration calculation scheme could be further automated. It would be a straight-forward 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 data.

### *B. Digital Beamforming*

The DBF FPGA processing is done in two parts. First, the calibration weights are applied and second, the 8-channels are beamformed through a matrix multiplication.

The calibration correction procedure is shown in Fig. 5. Given a current calibration set (comprised of 8 amplitude weights and 8 time delays) it is a straightforward though non-trivial process to apply it in real-time to streaming data arriving from the A/Ds. Eight of these blocks are declared and one block is applied to each channel. Each correction block consists of a 1024 point FFT (shared with the 1024 point FFT used above in the calibration block), a frequency stepping counter, and a sin/cos lookup table. Given 8-bit time domain data samples from the A/D, the FFT outputs 16-bit frequency domain data samples just as was done in the channel calibration block. These outputs are each multiplied by the $a_i$ for the current channel, thus correcting for any deviations in magnitude. Concurrently, a 10-bit counter (-512 to 511) is used to represent the range $-f_s/2$ to $f_s/2$. This value is multiplied by $\tau_i$ (time shift stored as a 10-bit fraction) and sent to a sin/cos lookup table. The table outputs 12-bit values which alter the real and imaginary amplitude corrected signals exiting the dual multiplier which follows the FFT. This operation corresponds to the simple transformation $Ax(t-\tau) => AX(\omega)exp(-j\omega\tau)$. Output samples are now completely compensated for amplitude and phase variations from the weights that were found during the calibration process.

The beamforming operation requires a large complex matrix multiplication given by

$$[\mathbf{T}]_{37x1024} = [\mathbf{A}]_{37x8}[\mathbf{F}]_{8x1024} \quad (1)$$

where **F** represents the FFT data of all 8 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, 37 scan angles ($-90^0$ to $+90^0$ in $5^0$ steps), and 8 channels.

It is apparent that 8 new complex samples are received every clock period from the data streaming out of the channel correction blocks. These 8 complex numbers comprise

TABLE I

FPGA OPERATIONS FOR DBF MATRIX COMPUTATION

| Scan | Step | x | +/- | A bits |
|---|---|---|---|---|
| 180 | 1 | 2896 | 1448 | 23168 |
| 180 | 2 | 1456 | 728 | 11648 |
| **180** | **5** | **592** | **296** | **4736** |
| 180 | 6 | 496 | 248 | 3968 |
| 120 | 1 | 1936 | 968 | 15488 |
| 120 | 2 | 976 | 488 | 7808 |
| 120 | 5 | 400 | 200 | 3200 |
| 120 | 6 | 336 | 168 | 2688 |
| 90 | 1 | 1456 | 728 | 11648 |
| 90 | 2 | 736 | 368 | 5888 |
| 90 | 5 | 304 | 152 | 2432 |
| 90 | 6 | 256 | 128 | 2048 |

one column of the **F** matrix. The clock period when they arrive at the beamforming block indexes the columns of the **F** matrix (frequency). Thus each clock period needs only to calculate the results for one column of the **T** matrix by using only one column from the **F** matrix. Unfortunately this calculation requires each value in the **A** matrix and therefore the entire matrix must be stored within the FPGA.

Separating the calculation into two FPGA's reduces the required computation by half but each FPGA still requires storage of the entire **A** matrix (not a big problem for reasonable numbers of scan angles in today's FPGA technology). Table I shows FPGA resources required for this calculation for varying beamforming scan angle configurations and bit precisions. All entries in Table I have 8-bit depth. Scan represents scan angle range in degrees (e.g. 180 is -90 to +90), Step represents scan angle step size in degrees, x represents total number of multiplications, +/- represent total number of additions and subtractions, and A represents the bit storage for matrix **A** from (1). Note that the bold row in Table I represents settings used collect results for this paper.

The FPGA calibration and DBF procedure was applied to the smart antenna system with measured data that was collected from a transmitted signal at broadside. Both calibration techniques were implemented. Fig. 6 shows the antenna pattern cut at broadside for the FPGA processed data. Again calibration method 2 compares more favorably with theory (sim.) than calibration method 1.

## V. CONCLUSIONS

This paper has shown that a smart antenna system with DBF and real-time FPGA processing is possible today for a moderate number of channels. This paper demonstrated the feasibility of an 8-channel system. Two calibration methods were presented and both were implemented with FPGA processing. After the calibration weights of the system are computed, the DBF is performed on collected signals from the smart antenna system via FPGAs. With advances in FPGA development and A/D converters, it will some day be practical and cost effective to build larger smart antennas with reconfigurable applications.

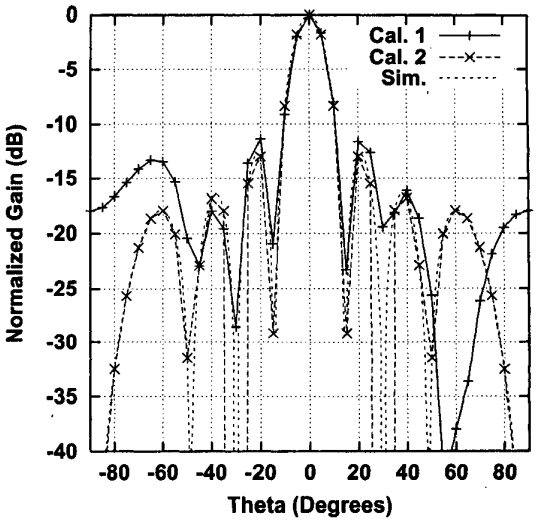


Fig. 6. Antenna pattern cut at broadside with FPGA processing.

## REFERENCES


[1] T. W. Nuteson and G. S. Mitchell, "Digital beamforming for smart antennas," *IEEE MTT-S Int. Microwave Symp. Dig.*, pp. 125-128, May 2001.

[2] Y. Hara, C. Ohno, M. Iwamoto, and N. Kondo, "A study of radiometric calibration of next generation spaceborne SAR," *IEEE International Geoscience and Remote Sensing*, vol. 3, pp. 1392-1394, Aug. 1997.

[3] S. Silverstein, "Application of orthogonal codes to the calibration of active phased array antennas for communication satellites," *IEEE Trans. Signal Processing*, vol. 45, no. 1, pp. 206-218, Jan. 1997.

[4] G. Brown, J. McClellan, and E. Holder, "A phased array calibration technique using eigenstructure methods," *IEEE International Radar Conference*, pp. 304-308, May 1990.

[5] J. Larocque, J. Litva, and J. Reilly, "Calibration of a smart antenna for carrying out vector channel sounding at 1.9 GHz," *Proceedings of the 8th Annual Virginia Tech. Symposium on Wireless Personal Communications*, pp. 293-302, June 1998.

[6] C. M. Simmonds and M. A. Beach, "Downlink calibration requirements for the TSUNAMI (II) adaptive antenna testbed," *Ninth IEEE Int. Symp. on Personal, Indoor, and Mobile Comm.*, vol. 3, pp. 1260-1264, Sept. 1998.

[7] C. Passman, F. Hickel, and T. Wixforth, "Investigation of a calibration concept for optimum performance of adaptive antenna systems," *48th IEEE Vehicular Tech. Conf.*, pp. 577-580, May 1998.

[8] J. H. Sinsky, "An internet controlled calibration system for TDMA smart antenna wireless base stations," *IEEE MTT-S Int. Microwave Symp. Dig.*, pp. 141-144, May 2001.

[9] R. G. Vaughan, N. L. Scott, and D. R. White, "The theory of bandpass sampling," *IEEE Trans. Signal Processing*, vol. 39, pp. 1973-1984, Sep. 1991.

[10] Xilinx, "Virtex-II 1.5V field-programmable gate arrays," *Xilinx Data Sheet*, pp. 1-4, April 2001.

[11] R. S. Adve and T. Sarkar, "Compensation for the effects of mutual coupling on direct data domain adaptive algorithms," *IEEE Trans. Antennas Propagat.*, vol. 48, no. 1, pp. 86-94, Jan. 2000.

[12] Xilinx, "Virtex-E 1.8V field-programmable gate arrays," *Xilinx Data Sheet*, pp. 1-4, April 2001.