Published in IET Microwaves, Antennas & Propagation
Received on 15th April 2009
Revised on 11th August 2009
doi: 10.1049/iet-map.2009.0151

In Special Issue on Selected Papers from Mosharaka International Conference on Communications, Propagation and Electronics (MIC-CPE 2009)

ISSN 1751-8725

Digital signal processor against field programmable gate array implementations of space–code correlator beamformer for smart antennas

S. Dikmese[1,2] *A. Kavak*[2] *K. Kucuk*[3] *S. Sahin*[2] *A. Tangel*[4] *H. Dincer*[4]

[1]*Department of Communication Engineering, Tampere University of Technology, Tampere, Finland*
[2]*Department of Computer Engineering, Wireless Communication and Information Systems Research Center, Kocaeli University, Kocaeli, Turkey*
[3]*Department of Electronic and Computer Education, Kocaeli University, Kocaeli, Turkey*
[4]*Department of Electronic and Communication Engineering, Kocaeli University, Kocaeli, Turkey*
E-mail: sener.dikmese@kocaeli.edu.tr

Abstract: Software radio implementations of beamformers on programmable processors such as digital signal processor (DSP) and field programmable gate array (FPGA) still remain as a challenge for the integration of smart antennas into existing wireless base stations for 3G systems. This study presents the comparison of DSP- and FPGA-based implementations of space–code correlator (SCC) beamformer, which is practical to use in CDMA2000 systems. Implementation methodology is demonstrated and results regarding beamforming accuracy, weight vector computation time (execution time) and resource utilisation are presented. The SCC algorithm is implemented on Texas Instruments (TI) TMS320C6713 floating-point digital signal processors (DSPs) and Xilinx's VirtexIV family FPGA. In signal modelling, CDMA2000 reverse link format is employed. The results show that beamformer weights can be obtained within less than 10 ms via implementation on c6713 DSP with direction-of-arrival (DOA) search resolution of $\Delta\theta = 2^{\circ}$, whereas it can be achieved within less than 25 μs on VirtexIV FPGA for five-element uniform linear array (ULA). These results demonstrate that FPGA implementation achieves weight vector computation in much smaller time (nearly 500 times) as compared to DSP implementation in this study.

1 Introduction

Mobile communication systems have been growing rapidly and services offered by them are being widely used all around the world. Increasing traffic rates, limited system capacity and low coverage range of base stations are the factors to be considered in the design of these systems. To overcome these issues, software radio implementation, advanced antenna systems and adaptive signal processing techniques have been considered over last decade. A software defined radio (SDR) is defined as a reconfigurable radio, in which the functionality is described by software [1, 2]. In an SDR, which provides a flexible radio architecture that allows changing the radio functionality in real-time, the same hardware can be used to implement different processes at different times.

A smart antenna system (SAS) employing an antenna array at the base station with advanced signal processing techniques adaptively adjust its beam pattern according to channel propagation dynamics. SAS decreases system complexity, expands coverage and increases data rate by efficiently utilising the bandwidth [3, 4]. One of the difficulties integrating a SAS into wideband code division

multiple access (WCDMA) systems is the implementation of algorithms on programmable processors. Implementation of beamforming algorithms on programmable processors such as digital signal processors (DSPs), field programmable gate arrays (FPGAs) or special type of application-specific integrated circuits (ASICs) is a key point for upgrading such flexible base stations [5, 6]. DSPs can be considered as special purpose CPUs succeeding fast instruction sequences, such as shift, add and multiply. On the other hand, FPGAs with their re-programmable logic gates are more hardware-oriented devices, and also preferred for higher processing speeds.

Recently, many researchers have studied implementation of various beamforming algorithms using different programmable processors. In [7], least mean square (LMS) and recursive least square (RLS) algorithms as a beamformer for WCDMA were implemented using Texas Instruments' (TI) C6211 DSP processor. In [8], implementation of normalised least mean squared (NLMS) beamformer was performed on TI C6203 DSP using two DSPs for physical layer and media access control (MAC) layer. In [9], a normalised constant modulus algorithm (NCMA) was implemented using Xilinx's SPARTAN II FPGA to study digital beamforming capability of an FPGA. In [10], a beamformer system consisting of an eight-element antenna array, eight TI C6701 DSP processors and eight co-processors was implemented using Xilinx's XCV400E FPGA technology.

In this paper, we extend our previous works [11–13] on DSP and FPGA implementations for wireless environments. We specifically focus on implementation of a smart antenna algorithm that we have developed earlier and referred to as space–code correlator algorithm (SCC) using TI floating-point DSP (C6713 DSK) [13] and Xilinx's VirtexIV FPGA. Signal received from the antenna array is assumed to be transmitted in CDMA2000 format [14]. The advantage of SCC algorithm is that unlike other adaptive algorithms such as LMS and constant modulus (CM) [15], it does not need any learning parameter and also its weight vector computation time is not affected by multipath propagation conditions [11].

The remainder of the paper is organised as follows. SCC algorithm is described in Section 2. Implementation methodology based on DSP and FPGA is presented in Section 3. Setup of implementation is explained in Section 4. Results pertaining resource utilisation, weight vector computation time, effect of direction-of-arrival (DOA) search resolution, effect of signal-to-noise ratio (SNR) variation and antenna configuration are presented in Section 5. Finally, the concluding remarks are given in Section 6.

2 Description of SCC algorithm

The SCC algorithm whose implementation on DSP and FPGA to be presented in this paper was also discussed in [13, 16]. It is based on performing code correlation with desired user's code and then spatial correlation of despread signal with predetermined array response vectors in the reverse link search table. However, we herein briefly describe this algorithm. The transmitted signal $s(t)$ from the mobile is exposed to multipath propagation environment, which induces complex path attenuation $\alpha_{i,f} = \beta_{i,f} e^{j\phi_{i,f}}$ and time delay $\tau_{i,f}$ on the transmit signal. Let f and F denote the multipath index and number of multipaths, respectively, from the desired mobile to the base station. The received signal at the input of an antenna array is given as

$$\boldsymbol{x}(t) = \underbrace{\sum_{f=1}^{F_1} \alpha_{1,f} s_1(t-\tau_{1,f}) \boldsymbol{a}(\theta_{1,f})}_{\text{desired}} + \underbrace{\sum_{i=2}^{I} \sum_{f=1}^{F_i} \alpha_{i,f} s_i(t-\tau_{i,f}) \boldsymbol{a}(\theta_{i,f})}_{\text{interference}} + \boldsymbol{n}(t) \quad (1)$$

where $\boldsymbol{a}(\theta_{1,f})$ is the antenna array response vector and $\boldsymbol{n}(t)$ is additive white Gaussian noise (AWGN) term, i is the interference index and I is the total number of interference.

SCC algorithm has two parts as code correlator and space correlator. In code correlator stage, received signal $\boldsymbol{x}(t)$ is despread by the code $c_1(t)$ of desired user to be obtain pth multipath

$$\boldsymbol{Z}_p(l) = \frac{1}{T_w} \int_{(l-1)T_w+\tau_{1',p}}^{lT_w+\tau_{1,p}} \boldsymbol{x}(t) c_1^*(t-\tau_{1,p}) \mathrm{d}t \quad (2)$$

where T_w is the symbol period and $\tau_{1,p}$ is the multipath delay for the pth path of the desired user. If baseband signal is sampled at chip instants (T_c), and the pulse shaping waveform is chosen as rectangular function with unit amplitude, then above equation can be written as to obtain post correlation signal vector

$$\boldsymbol{Z}_p(l) = \frac{1}{L} \sum_{m=0}^{L-1} c_1^*(l; m) \boldsymbol{x}(mT_c + \tau_{1,p}) \quad p = 1, \ldots, F \quad (3)$$

In spatial correlator part, correlation of $\boldsymbol{Z}_p(l)$ and array response vectors $\boldsymbol{a}(\theta)$ is carried out. The scope $[0^\circ, 180^\circ]$ is separated into K DOAs, which are divided by search resolution $\Delta\theta^\circ = 180^\circ/K$. Thus, K complex-valued steering vector with dimension $M \times 1$ $\boldsymbol{a}(\theta)$ is necessary to save in reverse link table. The output of space correlator tries to find the dot product of the code correlator output and the array response vector to find maximum DOA corresponding to its peak for each path

$$\hat{\theta}_p = \arg\max |\boldsymbol{a}(\theta)^{\mathrm{H}} \boldsymbol{Z}_p(l)|^2 \quad (4)$$

The estimated DOA can be applied for both uplink and downlink beamforming as long as user position changes

IET Microw. Antennas Propag., 2010, Vol. 4, Iss. 5, pp. 593–599
doi: 10.1049/iet-map.2009.0151

very slowly during several symbol periods. As can be observed above, the complexity and accuracy of the SCC beamformer depends on the number of multipath, correlation level of multipath, number of antenna elements in the array and angular resolution of the DOA range to be scanned.

3 DSP and FPGA implementation

3.1 DSP implementation

The implementation for the SCC algorithm on DSP was presented in our previous study [16]. Hence, we suggest reader to refer that paper for detailed information on DSP implementation. TI's C67x family DSPs were used in the implementation, which use some specific instructions which are 32-bit integer multiply, double word load and floating-point operations. Consequently, we used single precision of floating-point operands to code the algorithms. TMS320C67x DSPs use high-performance, advanced VelociTI very-long-instruction-word (VLIW) architecture [17–20], which enables multichannel and multifunction processing. The C67x processor consists of three main parts: CPU, peripherals and memory. Eight functional units operate in parallel, with two similar sets of the four functional units. The functional units communicate using a cross path between two register files, each of which contains 16 registers with 32-bit width. The 256-bit-length program memory fetches eight 32-bit instructions every single cycle.

3.2 FPGA implementation

Virtex FPGAs have an array of configurable logic blocks (CLBs) that are encircled by a ring of input/outputs blocks (IOBs). Block RAMs (BRAMs) are placed on the two sides of the FPGA. The CLBs are the main building blocks that consist of logic elements such as gates, flip flops and wiring for connectivity. Any CLB has two slices as an input multiplexer and an output multiplexer.

We have used a very high speed integrated circuit hardware description language (VHDL) library that we have previously designed for floating-point addition fp_add and floating-point multiplication fp_mul [21, 22].

Owing to code correlator and space correlator parts of SCC algorithm and limited size of FPGA, it was not feasible to compute weights of each antenna element in parallel fashion in SCC algorithm. Serial implementation is not as efficient as parallel implementation in terms of weight computation time, but we try to minimise the gap between the two by optimising our implementation of arithmetic blocks on FPGA. The operations of these units are managed by control unit. The 32-bit floating-point format requires too high process load, so we have to use 16-bit floating-point format (half IEEE754 floating-point format) to implement on VirtexIV FPGA.

The SCC implementation architecture for an FPGA is shown in Fig. 1. For the ease of understanding, we provide the explanation of implementation blocks as described in [13]. Implementation blocks on an FPGA are composed of six entities named as Main, Search Table, Space Correlator, Code Correlator, Received Signals and Abs entities.

Received signal entity: This entity is used as a buffer. The received signal entity provides an interface for incoming signals. Generating 1.2288 mega chips per second (MCPS) signal corresponds to data length of 24 576 sizes of data for 20 ms duration. Size of samples which is $S = 768$ is saved in the internal RAM of Virtex IV, while 0.625 ms duration is considered as the part of input signal. If number of antenna elements are taken as $M = 5$, the table size of RAM in the FPGA must be $M \times S \times 2$ because of real and imaginary parts of a signal sample. Values of received signal from antenna which has half IEEE 754 floating-point format is updated in this entity for every iteration.

Code correlator entity: Code correlator process is implemented for 64 complex multiplications (in multiplier entity, adder) and additions in total of $n = 60$ steps. Complexity of code correlator spends too much memory space on the FPGA. Hence, a sub-module which has eight complex multiplications at a time was implemented. The output of the code correlator entity is $M \times N$ complex-valued matrix whose elements are input to the space correlator entity.

Space correlator entity: The received $M \times N$ complex-valued data from code correlator entity is replaced by $M \times N$ size array. This entity provides correlation of $1 \times M$ complex-valued array response vector with the $M \times N$ size data. In each step $\theta_{n,j}$, we find $1 \times N$ sized complex data that corresponds to spatial correlation result.

Search table entity: This entity saves total of θ_n spatial angles which corresponds to array response vector. Hence, we require a table of size $\theta_n \times M$. In this implementation, θ_n is equal to 90 for $\Delta = 2^\circ$ resolution.

Abs entity: This entity computes absolute value of $1 \times N$ sized complex-valued data from space correlator entity for the each spatial angle in each step.

Main entity: This entity controls all entities in this implementation. It saves the received data X_r from space correlator entity in a Look-up table. Peak finder entity performs the largest value.

We can consider adder entity and multiplier entity as sub-entity.

Adder entity: In the complex adder entity, floating-point operations referred as functions are performed. In this entity, the addition function is called four times in order to perform a complex addition operation.

Multiplier entity: In the complex multiplier entity, floating-point operations referred as functions are performed. In the

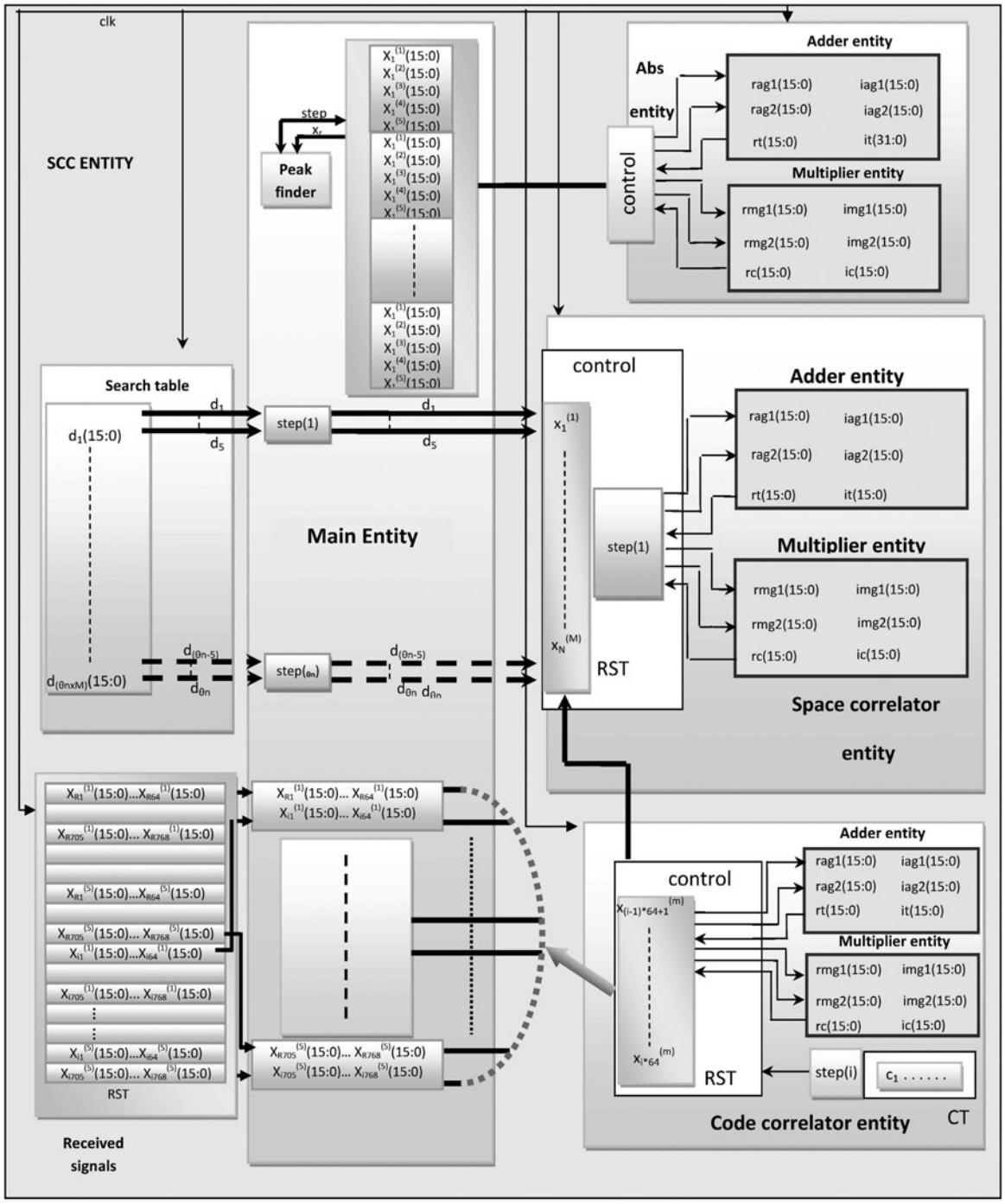

Figure 1 *Implementation architecture of SCC algorithm on an FPGA [13]*

complex multiplier entity, addition and multiplication functions are called two times and four times, respectively. Multiplier entity multiplies signal values from each antenna elements, which are received by main entity, with complex weight vectors stored in the RAM entity. These results are used for error correction in the next step.

4 Input signal parameters

We consider a five-element antenna array with uniform linear array (ULA) as receiving antenna. In signal modelling, a simple wireless channel model having a direct path and a multipath component for the desired signal and an interference signal is considered. DOA of the direct path ($\theta_{1,1}$) is fixed at 32°, whereas DOA of the multipath ($\theta_{1,2}$) and the interference signal ($\theta_{2,1}$) is changing randomly (with uniform distribution) from one simulation run to another. The amplitude (β) and phase (ϕ) components of multipath fading parameters are Rayleigh and uniform random variables, respectively. Multipath signal power level is set to 5, 10 and 15 dB below the direct path signal for testing the performance of the algorithm. Interference

IET Microw. Antennas Propag., 2010, Vol. 4, Iss. 5, pp. 593–599

doi: 10.1049/iet-map.2009.0151

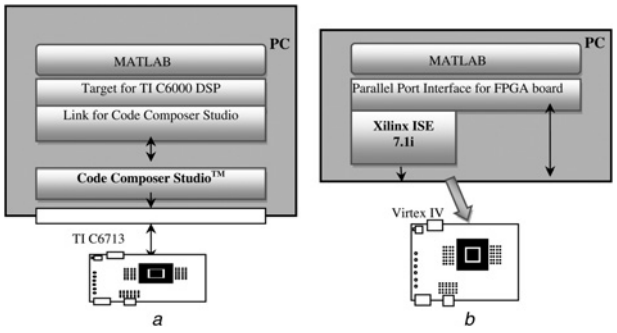

Figure 2 *Configuration of defined hardware structure of the*

a DSP system
b FPGA system

signal is 10 dB below the direct path of desired signal. In Fig. 2, hardware settings for DSP and FPGA implementations are depicted. All the signal parameters and signal samples using these parameters are generated in Matlab, and then loaded into relevant board for simulations.

5 Implementation results

We first test the performances of DSP and FPGA implemented beamformers' in terms of their beamforming accuracy. In the context here, beamforming accuracy is measured via DOA of the spatial spectrum peak. If this DOA is close enough to DOA of direct path of desired user's signal, we consider high beamforming accuracy. In other words, beamforming accuracy shows how closely beamformer's spatial spectrum pinpoints in the direction of desired user. Fig. 3 shows representative spatial spectrum results for DSP and FPGA implementations. Since the peak spectrum points nearly 32°, which is the direction of desired user, we obtain high beamforming accuracy for both DSP and FPGA implemented SCC beamformers.

The results regarding execution times for DSP and FPGA for DOA search resolution $\Delta\theta = 2^\circ$ are shown in Table 1. In DSP, weight vector computation or execution time in DSP implementation is determined via code composer studio (CCS) profiler with the clock cycles. The profiler calculates the program execution and presents the spent time. In FPGA, code correlator part is implemented as 64 complex multiplications and additions in 60 steps. Every step in this part spends 701 clock cycles that corresponds to 0.1402×10^{-3} ms because of 500 MHz clock frequency. Space correlator is calculated for 90 different angles for search resolution of $\Delta\theta = 2^\circ$ that corresponds to $0.1534 \times 10^{-3} \times 90$ ms. Finally, total execution time can be calculated as $0.1402 \times 10^{-3} \times 60 + 0.1534 \times 10^{-3} \times 90 = 0.022$ ms. Execution time of SCC algorithm on DSP and FPGA is obtained under different multipath channel conditions with random DOA and fading coefficient of multipath. Implementations on the same processor are repeated 100 times and execution time results are averaged to take into account these random variations in the wireless

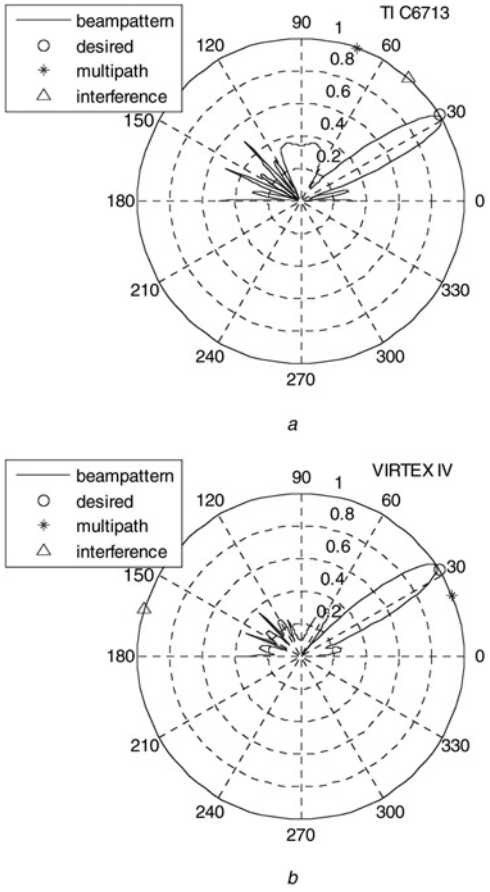

Figure 3 *Representative beamforming spectrum of the SCC beamformer implemented on*

a C6713 DSP
b VirtexIV

channel. Clearly, there is a significant improvement in execution time, when the same algorithm is implemented on the FPGA. FPGA implementation provides at least 500 times faster execution time compared to DSP.

Although DSP and FPGA have different hardware structures, it is useful to examine resource utilisations in the algorithm implementation. Owing to limited resources on FPGA (flip flops, LUTs, slices), we were able to implement SCC algorithm with $\Delta\theta = 2^\circ$ using 16-bit floating-point format (half floating point) on VirtexIV. DSP and FPGA resource utilisation is given in Table 2. In terms of percent resource utilisation of their own, DSP requires less resource than FPGA.

Table 1 Execution time for SCC algorithm

	VirtexIV (500 MHz)	C6713 DSP
code correlator	$0.1402 \times 60 \times 10^{-3}$ ms	2.9 ms
space correlator	$0.1534 \times 90 \times 10^{-3}$ ms	6.38 ms
total time	0.022 ms	9.28 ms

Table 2 SCC algorithm synthesis results

	Virtex4(xc4vlx60)	TI C6713
Number of slice flip flops	10,727 of 53,248 (20.14%)	—
Number of four input LUTs	43,560 of 53,248 (81.80%)	—
Number of occupied slices	26,622 of 26,624 (99.99%)	—
internal program memory		15.6%
external data memory		15%

Table 3 DOA estimation of SCC algorithm

Device	Virtex4(xc4vlx60)			TI C6713		
SNR, dB	5 dB	10 dB	15 dB	5 dB	10 dB	15 dB
average DOA estimation error	6	5	4	4	3	2.5

SCC algorithm's execution time is not affected by the change in multipath DOA ($\theta_{i,f}$), fading level ($\alpha_{i,f}$) and antenna array topology. However, SNR level is crucial in separating direct path and multipath DOAs. In Table 3, mean values of DOA estimation errors obtained from implementation of 100 repetitions are summarised for ULA topology. DSP implementation leads to slightly smaller DOA estimation errors than FPGA for all SNR conditions.

6 Conclusions

We have presented a comparative study of a beamformer implementation on DSP (TI C6713) and FPGA (Xilinx Virtex IV). As a beamformer, our previously developed SCC algorithm, which is well suited for 3G CDMA applications, was selected. In signal modelling, CDMA2000 reverse link channel and five-element ULA were considered. The performance evaulation of the implemented SCC algorithm on the DSP and the FPGA was made in terms of beamforming accuracy, execution time, resource utilisation and DOA estimation error. Both the DSP and FPGA were able to provide a weight vector that can track the desired user direction. In terms of execution time (weight vector computation time), the FPGA implementation resulted in much faster execution time (500 times faster) when compared to the DSP implementation. Hence, as expected FPGAs can be used to reduce the execution time. The implementation of the SCC algorithm on Virtex4 FPGA with 16-bit floating point (half floating point) used up to approximately 99% of physical resources; on the other hand, DSP required only 30% of its memory resources. In a further study, optimisation of the implementation can be made by a hybrid implementation method which requires the use of both an FPGA and a DSP.

7 Acknowledgment

This work was partially supported by Kocaeli University Scientific Researches Divisions under the project number of KOU-BAP 2005/58.

8 References

[1] PATTI J.J., HUSNAY R.M., PINTAR J.: 'A smart software radio: concept development and demonstration', *IEEE J. Sel. Areas Commun.*, 1999, **17**, (4), pp. 631–649

[2] BURNS P.: 'Software defined radio for 3G' (Artech House, MA, 2002), pp. 221–237

[3] RAPPAPORT T.S., LIBERTI J.C.: 'Smart antennas for wireless communication' (Prentice-Hall, NJ, 1999)

[4] GODARA L.: 'Application of antenna arrays to mobile communications, Part II: beamforming and direction-of-arrival considerations', *Proc. IEEE*, 1997, **85**, (8), pp. 1195–1245

[5] Texas Instruments: TMS320C6711, TMS320C6711B, TMS320C6711C Floating-Point Digital Signal Processors. Literature No. SPRS088D, Texas, USA, 2002

[6] XILINX : 'Virtex-II 1.5V field-programmable gate arrays', *Xilinx Data Sheet*, 2001, DS 031-1(V1.7) pp. 1–4

[7] XI Y.H., YANG F.: 'RLS-based blind adaptive beamforming algorithm for antenna array in CDMA systems'. Proc. Int. Conf. Information Acquisition, July 2005, p. 6

[8] MARTINEZ R., GARCIA L., DE HARO L., CALVO M.: 'A DSP-based implementation of adaptive algorithms for a W-CDMA reverse link beamformer'. Proc. IEEE Rawcon'02, August 2002, pp. 141–144

[9] EIREINER T., MULLER T., LUY J.-F., OWENS F.: 'Implementation of a smart antenna system with an improved NCMA algorithm', *IEEE MTT-S Int. Microw. Symp. Dig.*, 2003, **3**, pp. 1529–1532

[10] BUCCI G., COLAMONICO A., DONATI M., PICCIRIELLO A., SPAGNOLINI U.: 'Smart antenna BTS based on software radio technique for GSM/DCS system'. Proc. IEEE VTC'00, May 2000, vol. 2, pp. 1225–1229

[11] KUCUK K., KARAKOC M., KAVAK A., YIGIT H.: 'Design and hardware implementation of a novel smart antenna

doi: 10.1049/iet-map.2009.0151

algorithm using TI DSPs'. Proc. of IEEE ISWCS'05, September 2005, pp. 596–600

[12] SAHIN S., DIKMESE S., KUCUK K., KAVAK A.: 'A comparative study of antenna array algorithm implementations using FPGA and DSP for CDMA2000'. Third Int. Symp. Wireless Communication Systems (ISWCS 2006), September 2006, pp. 714–718

[13] DIKMESE S., KAVAK A., KUCUK K., SAHIN S., TANGEL A.: 'FPGA based implementation and comparison of beamformers for CDMA2000'. *Wirel. Pers. Commun.*, 2009, DOI 10.1007/s11277-009-9855-4

[14] TIA/EIA Interim Standard: 'Physical layer standard for CDMA2000 spread spectrum systems', TIA/EIA/S-2000-2

[15] VEEN A.J., PAULRAJ A.: 'An analytical constant modulus algorithm', *IEEE Trans. Signal Process.*, 1996, **44**, (5), pp. 1136–1155

[16] KUCUK K., KAVAK A., KARAKOC M., YIGIT H., OZDEMIR C.: 'A practical space–code correlator receiver for DSP based software radio implementation in CDMA2000', *Wirel. Pers. Commun.*, 2009, **49**, (2), pp. 245–261

[17] SESHAN N.: 'High VelociTI processing [Texas Instruments VLIW DSP architecture]', *IEEE Signal Process. Mag.*, 1998, **15**, (2), pp. 86–101, 117

[18] Texas Instruments: TMS320C6201/6701 Evaluation Module Technical Reference. Literature No. SPRU305, Texas, USA, 1998

[19] Texas Instruments: TMS320C6711, TMS320C6711B, TMS320C6711C Floating-Point Digital Signal Processors. Literature No. SPRS088D, Texas, USA, 2002

[20] Spectrum Digital Inc.: TMS320C6713 DSK Technical Reference. 506735-0001 Rev. A, Stafford, TX, USA, 2003

[21] LIGON W.B., MCMILLAN S., MONN G., SCHOONOVER K., STIVERS F., UNDERWOOD K.D.: 'A re-evaluation of the practicality of floating point operations on FPGAs'. Proc. IEEE Symp. Field-Programmable Custom Computing Machines, April 1998, pp. 206–215

[22] SAHIN S., KAVAK A., BECERIKLI Y., DEMIRAY H.E.: 'Implementation of floating point arithmetic using an FPGA', in TAS K., MACHADO J.A., BALEANU D. (EDS.): Mathematical methods in engineering (Springer Book, Netherlands, 2007)