

Design and FPGA implementation of module for space multiplexing in multi-user MIMO system

Abstract. MIMO (multiple in multiple out) antenna system draw attention in the recent years due to its potential for achieving high data rates. In this work novel DSP algorithm and FPGA implementation will be presented that allows transmitting M data streams to M receiving antennas from N transmitting antennas ($M \leq N$) into the same frequency with total interference suppression while maximizing channel gain for each data stream at the same time.

Streszczenie. W artykule zaprezentowano nowy algorytm DSP z implementacją w FPGA umożliwiającą transmisję strumienia M danych do M anteny odbiornika z anteny przesyłowej N ($M \leq N$) z tą samą częstotliwością z tłumieniem zakłóceń – przy maksymalizacji wzmocnienia dla każdego strumienia danych w kanale. (Projekt i implementacja FPGA modułu do przestrzennego multipleksowania w systemie MIMO w wieloma użytkownikami)

Keywords: FPGA, VHDL, MIMO systems, Gram-Schmidt process, spectral efficiency

Słowa kluczowe: FPGA, system MIMO, process Grama-Schmidta

Introduction

In the past decade numerous DSP algorithms that utilise MIMO system potential for achieving high data rates were presented [1-3]. Many of these algorithms are using multiuser interference cancellation (MUIC) techniques which are usually based on SVD (singular value decomposition) method. Methods for SVD calculation, described in [4], are complex iterative algorithms with also complex FPGA implementation [5]. In some papers methods based on Gram Schmidt orthogonalization (GSO) [6] or Cholesky factorisation [7] are used for MUIC for achieving smaller complexity.

But none of the above papers are dealing with implementation of presented DSP algorithms. In this paper new DSP algorithm (Successive interference cancellation with GSO – SICwGSO) based entirely on GSO is presented and realized in Xilinx FPGA chip.

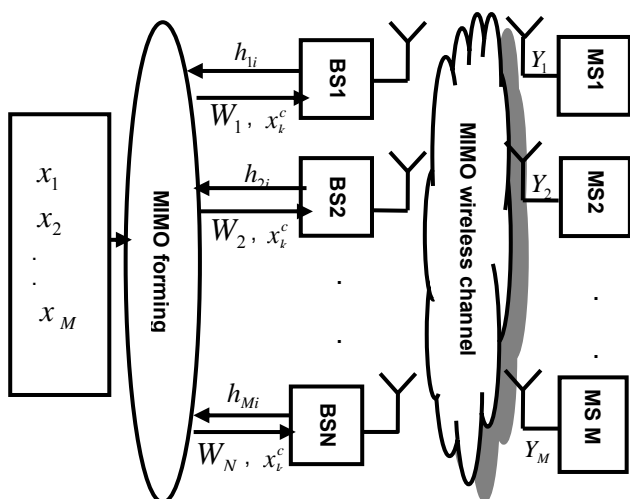


Fig. 1 System model

SICwGSO is described in details, FPGA implementation is presented and its performance through hardware co-simulation is evaluated. Then complexity of this algorithm is compared to algorithm in [8] (Block diagonalisation for throughput maximization - BDfTM), which uses SVD method for MUIC. This BDfTM algorithm is used as a reference since in many papers it is a reference for comparison [9][10]. It is shown that SICwGSO is much less complex than BDfTM and can be easily realized with simple combinational network in FPGA. Last section is the conclusion of this paper.

System model

In Fig. 1 system model is presented where symbols x_1, x_2, \dots, x_M are transmitted to users MS1 to MSM from base stations BS1 to BSN in the same time slot and using same frequency band. Base stations (BS) N transmitting antennas and M users receiving antennas forms $N \times M$ MIMO system. Each BS receives, from MIMO forming block, its transmitting/receiving coefficients $W_1 \dots W_N$ and modulated symbol values x_k^c which are calculated in MIMO forming block, based on knowledge of $N \times M$ MIMO channel values.

It is assumed that these channel values are known at BS's.

System equations

For system in Fig. (1) system equations are:

$$\begin{aligned}
 Y_1 &= W_{R1} \times H_1 \times X_{tr} + W_{R1} \times N_{S1} \\
 &\vdots \\
 Y_i &= W_{Ri} \times H_i \times X_{tr} + W_{Ri} \times N_{Si} \\
 &\vdots \\
 Y_M &= W_{RM} \times H_M \times X_{tr} + W_{RM} \times N_{SM} \\
 X_{tr} &= (W_{T1} \cdot x_1 + \dots + W_{Ti} \cdot x_i + \dots + W_{TM} \cdot x_M)
 \end{aligned}
 \tag{1}$$

where ' \times ' is a vector multiplication and ' \cdot ' is scalar multiplication. Each symbol x_i is multiplied with transmitting vector W_{Ti} and is transmitted from all transmitting antennas. Values H_i presents channel coefficients from all transmitting antennas to user MS i . Value W_{Ri} is receiving coefficient for user MS i and N_{Si} represents white noise sample for user MS i . Equations from (1) can be presented as:

$$\begin{aligned}
 Y_i &= B_i + W_{Ri} \times H_i \times W_{Ti} \cdot x_i + F_i + W_{Ri} \times N_{Si} \\
 B_i &= W_{Ri} \times H_i \times (W_{T1} \cdot x_1 + \dots + W_{T(i-1)} \cdot x_{i-1}) \\
 F_i &= W_{Ri} \times H_i \times (W_{T(i+1)} \cdot x_{i+1} + \dots + W_{TM} \cdot x_M)
 \end{aligned}
 \tag{2}$$

where values B_i and F_i represents *backward* and *forward* interference. In this paper SICwGSO algorithm for suppressing backward interference and its implementation in FPGA Xilinx Virtex-4 chip will be presented. Forward interference suppression method will be briefly described.

SICwGSO algorithm

Equations in (1) and (2), with assumption that $F_i=0$, are:

$$\begin{aligned}
 Y_1 &= W_{R1} \times H_1 \times W_{T1} \cdot x_1 + W_{R1} \times N_{S1} \\
 Y_2 &= W_{R2} \times H_2 \times W_{T1} \cdot x_1 + W_{R2} \times H_2 \times W_{T2} \cdot x_2 + W_{R2} \times N_{S2} \\
 Y_3 &= W_{R3} \times H_3 \times W_{T1} \cdot x_1 + W_{R3} \times H_3 \times W_{T2} \cdot x_2 + W_{R3} \times H_3 \times W_{T3} \cdot x_3 + W_{R3} \times N_{S3} \\
 &\vdots \\
 Y_M &= W_{RM} \times H_M \times W_{T1} \cdot x_1 + W_{RM} \times H_M \times W_{T2} \cdot x_2 + \dots + W_{RM} \times H_M \times W_{T(M-1)} \cdot x_{M-1} + \\
 &\quad + W_{RM} \times H_M \times W_{TM} \cdot x_M + W_{RM} \times N_{SM} \dots
 \end{aligned}
 \tag{3}$$

I_1 I_2 I_{M-1}

where values in sets $I_1 \dots I_{M-1}$ represents backward interference. Gram Schmidt method will be used for backward interference suppression.

With Gram Schmidt method [11] orthogonal space for any vector can be calculated. Equations for calculating orthogonal space using Gram Schmidt method are:

$$\begin{aligned}
 R_1 &= V_1 \\
 R_2 &= V_2 - R_1 \cdot (R_1 \times V_2) / \|R_1\| \\
 R_3 &= V_3 - R_1 \cdot (R_1 \times V_3) / \|R_1\| - R_2 \cdot (R_2 \times V_3) / \|R_2\| \\
 &\vdots \\
 R_N &= V_N - R_1 \cdot (R_1 \times V_N) / \|R_1\| - \dots - R_{N-1} \cdot (R_{N-1} \times V_N) / \|R_{N-1}\|
 \end{aligned}
 \tag{4}$$

where $\|R_i\|$ is a module of R_i . Vectors $R_1 \dots R_N$ represents orthogonal space and vectors $V_1 \dots V_N$ represents input vectors for calculating $R_1 \dots R_N$.

If we want to suppress backward interference I_j then $V_1 = H_{j+1}$ and vectors $V_2 \dots V_N$ can have any value. Vectors $V_2 \dots V_N$ represents degrees of freedom which can be used for optimization of FPGA implementation. Now matrix can be formed as:

$$W_{j(j+1)} = [R_2^T \dots R_N^T].
 \tag{5}$$

This matrix is orthogonal on $V_1 = H_{j+1}$. Then this process will be repeated for matrix $H_{j+2}^m = H_{j+2} \times W_{j(j+1)}$ with result $W_{j(j+2)}$, then for matrix $H_{j+3}^m = H_{j+3} \times W_{j(j+1)} \times W_{j(j+2)}$ with result $W_{j(j+3)}$ and so on until matrix W_{jM} is calculated for

$H_M^m = H_M \times W_{j(j+1)} \times W_{j(j+2)} \times \dots \times W_{j(M-1)}$. The resulting matrix is:

$$W_j = W_{j(j+1)} \times W_{j(j+2)} \times \dots \times W_{jM}.
 \tag{6}$$

Matrix W_j , which suppress backward interference $I_j, j \in \{1, \dots, M-1\}$, has dimension $N \times j$ and is orthogonal to $H_{j+1} \dots H_M$. Transmitting vectors are calculated as:

$$\begin{aligned}
 W_{Ti} &= (W_i \times D_i) \quad i \in \{1, \dots, M-1\} \\
 W_{TM} &= H_M
 \end{aligned}
 \tag{7}$$

where D_i is vector that will be calculated to maximize transfer function $H_i \times W_{Ti}$ for symbol x_i as:

$$\begin{aligned}
 H_i \times W_{Ti} &= H_i \times (W_i \times D_i) = (H_i \times W_i) \times D_i \\
 D_i &= (H_i \times W_i)^T
 \end{aligned}
 \tag{8}$$

Since transfer functions $10 \log(H_i \times W_{Ti}), i \in \{1, \dots, M\}$ (value in dB) have different values *water filling rule* will be used to equalize transfer functions values to the mean value G where:

$$\begin{aligned}
 10 \log(G) &= \frac{1}{M} \sum_{i=1}^M 10 \log(H_i \times W_{Ti}) \\
 G &= \sqrt[M]{\prod_{i=1}^M (H_i \times W_{Ti})}
 \end{aligned}
 \tag{9}$$

as it is desired that all users experience same QOS (quality of service). According to *water filling rule* transmitting vector values W_{Ti} are calculated and its values normalized as:

$$\begin{aligned}
 c_i &= G / (H_i \times W_{Ti}) \\
 W_{Ti} &= W_{Ti} \cdot c_i \\
 W_{Ti} &= W_{Ti} / \sqrt{\|W_{Ti}\|}
 \end{aligned}
 \tag{10}$$

Received coefficients W_{Ri} are calculated as:

$$W_{Ri} = 1 / (H_i \times W_{Ti}) = 1 / G.
 \tag{11}$$

Forward interference suppression

From (2) the forward interference can be presented as:

$$\begin{aligned}
 F_M &= 0 \\
 F_{M-1} &= W_{R(M-1)} \times H_{(M-1)} \times W_{TM} \cdot x_M^c \\
 &\vdots \\
 F_i &= W_{Ri} \times H_i \times (W_{T(i+1)} \cdot x_{i+1}^c + \dots + W_{TM} \cdot x_M^c) \\
 &\vdots \\
 F_1 &= W_{R1} \times H_1 \times (W_{T2} \cdot x_2^c + \dots + W_{TM} \cdot x_M^c)
 \end{aligned}
 \tag{12}$$

where x_k^c are the modified symbol values calculated to suppress forward interference. These modified values can be calculated as:

$$\begin{aligned}
 x_M^c &= x_M \\
 x_{M-1}^c &= x_{M-1} - F_{M-1} / (W_{R(M-1)} \times H_{(M-1)} \times W_{T(M-1)}) = x_{M-1} - F_{M-1}
 \end{aligned}
 \tag{13}$$

$$x_1^c = x_1 - F_1 / (W_{R1} \times H_1 \times W_{T1}) = x_1 - F_1$$

SICwGSO algorithm for 3x3 MIMO system

For 3x3 MIMO system equation (2) is:

$$Y_1 = W_{R1} \times H_1 \times W_{T1} \cdot x_1 + W_{R1} \times N_{S1}$$

$$Y_2 = W_{R2} \times H_2 \times W_{T1} \cdot x_1 + W_{R2} \times H_2 \times W_{T2} \cdot x_2 + W_{R2} \times N_{S2}$$

$$Y_3 = W_{R3} \times H_3 \times W_{T1} \cdot x_1 + W_{R3} \times H_3 \times W_{T2} \cdot x_2 + W_{R3} \times H_3 \times W_{T3} \cdot x_3 + W_{R3} \times N_{S3}$$

Backward interference values are $I_1 = \{W_{R2} \times H_2 \times W_{T1}, W_{R3} \times H_3 \times W_{T1}\}$ and $I_2 = \{W_{R3} \times H_3 \times W_{T2}\}$. We will design FPGA module, according to previously described SICwGSO algorithm, that will calculate W_{T1} and W_{T2} such that $I_1 = \{0, 0\}$ and $I_2 = \{0\}$.

According to Gram Schmidt method $W_{T11} = [R_1^T \ R_2^T]$ where:

$$\begin{aligned}
 R_1 &= V_1 \cdot \|H_3\| - H_3 \cdot (V_1 \times H_3) \\
 R_2 &= V_2 \cdot \|H_3\| \cdot \|R_1\| - H_3 \cdot (V_2 \times H_3) \cdot \|R_1\| - R_1 \cdot (V_2 \times R_1) \cdot \|H_3\|
 \end{aligned}
 \tag{15}$$

and $W_{T12} = [R_3^T]$ where:

$$\begin{aligned}
 R_3 &= V_3 \cdot \|H_2^m\| - H_2^m \cdot (V_3 \times H_2^m) \\
 H_2^m &= H_2 \times W_{T11}
 \end{aligned}
 \tag{16}$$

Then W_{T1} is calculated as:

$$W_{T1} = W_{T11} \times W_{T12} \tag{17}$$

Vector W_{T2} is calculated as:

$$W_{T2} = W_{T11} \times (H_2 \times W_{T11})^T \tag{18}$$

SICwGSO algorithm design for 3x3 MIMO system for FPGA in Xilinx ISE

In Xilinx ISE (Integrated System Environment) SICwGSO is developed in VHDL.

In calculation (15) value of V_1 is needed. This vector represents degrees of freedom which will be used to optimize FPGA design. Value of V_1 is chosen to be [2 2 1]. Choice of elements of this vector to the value of 2^S where S is an integer, simplifies FPGA design because for multiplying operation with V_1 shift module can be used instead of multiplier module. Multiplier module is much more complex and causes more propagation delay in FPGA design. Similar is valid for V_2 in (15) and V_3 in (16).

There is not enough space in this paper to present whole design so only the top level design is presented in Fig. 2

Inputs in FPGA module are channel values H_3 and H_2 and outputs are W_{T1} and W_{T2} . From Fig. 2 and also from (17) and (18) it can be seen that value of W_{T11}

(calculated by 'grsch1' block in Fig. 2) is used for calculating W_{T1} as well as W_{T2} , so complexity of FPGA realization is further reduced.

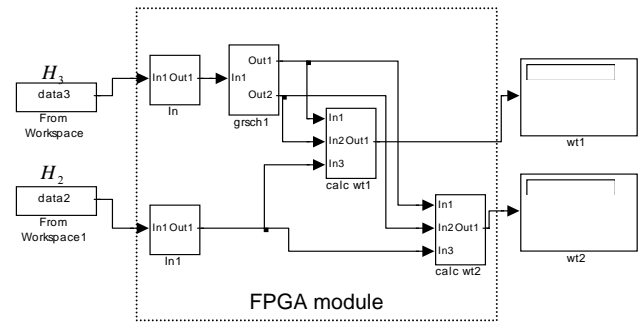


Fig. 2 Top level design

Hardware co-simulation

Design presented in Fig. 2, developed in VHDL in Xilinx ISE environment, is compiled and downloaded to ML402 development board (with XC4VSX35-FF668-10 FPGA Xilinx Virtex-4 chip) and it's performance is evaluated. Design in Fig. 2 is pure combinational, but due to propagation delay that result in negative clocking skew, it can't be compiled and memory elements (Z^{-1} blocks) must be added behind each multiplier block. Six clock intervals is now needed for design to be executed and propagational delay is roughly 1/6 of the value before. Clocking period (which is 33 ns) is now larger than largest propagation delay and design can be compiled and downloaded. Now design is ready for hardware co-simulation with Matlab program that will simulate BER (bit error rate) of the entire system without forward interference (described with equations in (14)) using transmitting vectors W_{T1} and W_{T2} which are calculated in FPGA Virtex-4 chip configured with design in Fig. [2]. Equations (9), (10) and (11) are not part of the FPGA design and are calculated in Matlab program. Resulting BER is presented in Fig. 3.

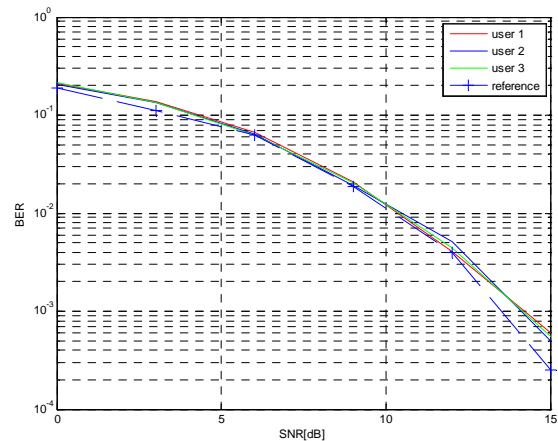


Fig. 3 BER for 3 users vs. reference BER

In Fig. 3 BER characteristics of 3 users are compared with reference characteristic without multiuser interference. It can be seen that, with calculation of transmitting vectors in FPGA chip, all users have BER characteristics like they are alone in the channel. In other words backward interference is almost totally suppressed and all users have the same quality of service which is the main goal of SICwGSO algorithm.

Complexity

Complexity of SICwGSO algorithm is calculated here in terms of required number of operations. Sum of equations in (19) represents required number of operation for SICwGSO algorithm for $N \times M$ MIMO system.

$$\sum_{i=1}^M \sum_{j=1}^i (N-j) \cdot (1+2+\dots+N-j) = \sum_{i=1}^M \sum_{j=1}^i (N-j)^2 \cdot (N-j+1)/2$$

$$(19) \sum_{i=1}^M \sum_{j=1}^i (N-j)^2 \cdot (N-j-1)$$

$$\sum_{i=1}^M \sum_{j=1}^i N \cdot (N-j) \cdot (N-j-1)$$

First equation represents number of operations required for Gram Schmidt method execution (equations in (4)). Second equation represents number of operations required for calculating modified transfer matrices (matrices H_j^m) and third equation represents number of operations required for calculating transmitting matrix (equations in (6)). Complexity of reference BDfTM algorithm can be calculated according to SVD complexity given in [12] as:

$$(20) \quad M \cdot (4 \cdot N + 8 \cdot N^2 + 9 \cdot N^3)$$

In Table 1 number of operation for these two methods are calculated when $N=M$.

Table 1. Complexity

$N=M$	2	3	4	5	6
SICwGSO	18	188	884	2840	7270
BDfTM	224	981	2880	6725	13536

It can be seen that SICwGSO has significantly lower number of operations especially for MIMO systems which are of practical interest (MIMO size less or equal to 4×4). Complexity results are also presented graphically in Fig. 4

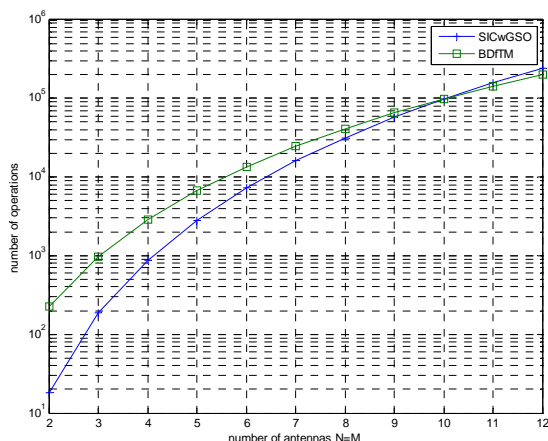


Fig. 4 Complexity comparison between SICwGSO and BDfTM

Lower complexity of SICwGSO algorithm means that less FPGA chip area is occupied. Also, since SICwGSO is non iterative algorithm, it produces less propagational delay than reference BDfTM algorithm which is iterative.

Conclusion

In this work novel DSP algorithm (Successive interference cancelation with GSO – SICwGSO) for MIMO system for achieving high spectral efficiency is presented. SICwGSO algorithm suppresses backward interference and is based on Gram Schmidt method which is simple to implement on FPGA. BDfTM (Block diagonalisation for

throughput maximization) algorithm for backward interference suppression is using SVD (singular value decomposition) which implementation on FPGA is very complex. Complexity of these two algorithms are compared in previous section. SICwGSO algorithm is much less complex than BDfTM algorithm especially for MIMO systems of practical interest (sizes less or equal to 4×4).

SICwGSO algorithm, for 3×3 MIMO system, is implemented on FPGA Xilinx Virtex-4 chip and it's performance evaluated through so called hardware co simulation. Simulation results, in Fig. 3, shows that BER (bit error rate) for all three users are practically the same as reference BER when there is no multiuser interference. This means that backward interference is almost totally suppressed which is the main goal of the SICwGSO algorithm.

Xilinx ISE tool is used for SICwGSO implementation on FPGA. Degrees of freedom that SICwGSO algorithm has, are used for optimization of FPGA implementation. Also, some implementation issues (like negative clocking skew) are presented and solution explained.

Main contribution of this work is that new SICwGSO algorithm and it's FPGA implementation is presented which is much less complex than reference BDfTM algorithm based on SVD method.

REFERENCES

- [1] Spencer Q.H., 'Capacity and downlink transmission algorithms for a multi-user MIMO channel', Conference on Signals, Systems and Computers 2002, pp. 1384 - 1388 vol.2
- [2] Quentin H. Spencer, A. Lee Swindlehurst, 'A hybrid approach to spatial multiplexing in multiuser MIMO downlinks', EURASIP Journal on Wireless Communications and Networking - Special issue on multiuser MIMO networks, Volume 2004 Issue 2, 15 December 2004, pp. 236-247
- [3] Liu, W., 'SVD-Assisted Multiuser Transmitter and Multiuser Detector Design for MIMO Systems', Vehicular Technology IEEE Transactions on, Volume 58, Feb. 2009, pp. 1016-1021
- [4] Alan Kaylon Cline, Inderit S. Dhillon, 'Computation of Singular Value Decomposition', The University of Texas at Austin, Chapter 45 of Handbook of Linear Algebra, edited by Leslie Hogben et al, CRC Press, 2006.
- [5] Weiwei Ma, 'An FPGA-Based Singular Value Decomposition Processor', Electrical and Computer Engineering, 2006. CCECE '06. Canadian Conference on, pp. 1047-1050
- [6] Gao, X., 'Low-complexity RBD precoding method for downlink of multi-user MIMO system', Electronic Letters, Volume 46, November 11 2010, pp. 1574-1576
- [7] Zukang Shen, 'Low complexity user selection algorithms for multiuser MIMO systems with block diagonalization', Signal Processing, IEEE Transactions on, Sept. 2006, pp. 3658-3663
- [8] Spencer, Q.H., 'Zero-forcing methods for downlink spatial multiplexing in multi-user MIMO channels', IEEE Transaction on Signal Processing, Volume 52, pp. 461-471, Feb. 2004.
- [9] Tarighat A., 'A multi user beamforming scheme for downlink MIMO channels based on maximizing signal-to-leakage ratios', Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on, Volume 3, pp. iii/1129 - iii/1132
- [10] Del Galdo, G. 'Comparison of zero-forcing methods for downlink spatial multiplexing in realistic multi-user MIMO channels', Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th, pp. 299-303 Vol. 1
- [11] Per-Olof Persson, 'Introduction to numerical methods', MIT 18.335: Introduction to Numerical Methods, lecture Fall 2007
- [12] Prof. Dr. Nassir Navab, '3D Computer Vision Script Draft v.000', appendix 9, TUM (Technische Universität München) lecture winter 2005/2006

Authors: Magister of science in Electrical Engineering Rade Dabetić e-mail: rade_dabetic@yahoo.com, Associate professor Lazar Saranovac e-mail: laza@el.etf.rs, University of Belgrade-School of Electrical Engineering, Department for Electronics, Bul. kralja Aleksandra 73 Belgrade, Serbia