

Implementação em FPGA de um decodificador MIMO-OFDM

K. Iguchi ^{a,c}, E.R. de Lima ^b, F.Angarita ^c, J.Valls ^c

Resumo – Este artigo descreve a implementação realizada em FPGA de um receptor MIMO-OFDM Zero Forcing com decomposição QR. O decodificador implementado utiliza o algoritmo CORDIC, e atende às especificações dos padrões HIPERLAN/2 e IEEE 802.11a.

Palavras-chave – MIMO-OFDM, FPGA, CORDIC

Abstract – This paper presents an implementation in FPGA of a MIMO-OFDM Zero Forcing decoder based on QR decomposition. The implemented decoder uses CORDIC in its algorithm, and works under the HIPERLAN/2 and IEEE802.11a standards specifications.

Keywords – MIMO-OFDM, FPGA, CORDIC

I. INTRODUÇÃO

Wireless Local Area Networks (WLANs) permitem conexões *wireless* de banda larga entre equipamentos eletrônicos. As normas Hiperlan/2 [1] e IEEE802.11a [2] suportam diferentes modos de transmissão, fornecendo altas taxas de dados, de até 54 Mbps na banda de 5GHz, dependendo das características do canal utilizado. Ambos se baseiam em Multiplexação Ortogonal por Divisão de Frequência (OFDM). Para combater efeitos de desvanecimento nas sub-portadoras OFDM causados por canais seletivos em frequência, se empregam codificação convolucional (FEC) e *interleaving* juntos. A codificação FEC usada se baseia em um codificador convolucional padrão de taxa 1/2.

Se pode combinar OFDM e *arrays* de antenas transmissoras e receptoras para aumentar a diversidade e/ou melhorar a capacidade do sistema em canais variáveis no tempo e seletivos em frequência, resultando em uma configuração múltiplas entradas – múltiplas saídas (MIMO) [3,4]. A Fig.1 abaixo ilustra um sistema de comunicação MIMO-OFDM:

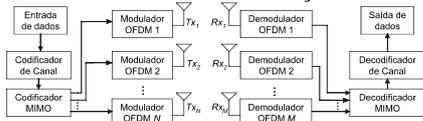


Fig.1: sistema $M \times N$ de transmissão e recepção MIMO-OFDM. O sistema analisado e apresentado neste artigo possui duas antenas de transmissão, Tx_1 e Tx_2 , e duas de recepção, Rx_1 e Rx_2 (sistema 2×2) e pode alcançar até 108Mbps na configuração de 64QAM com taxa de codificação 3/4.

II. QRD E O MÉTODO DE GIVENS

O decodificador MIMO-OFDM implementado está baseado na decomposição QR (QRD) da matriz de canal estimado H para a solução por mínimos quadrados¹ (LS – *Least Square*), através das *rotações de Givens* [5], um método que,

^a LME EPUSP, Lab. Microeletrônica da Escola Politécnica da Universidade de São Paulo, Brasil

^b Depto. Comunicaciones UPV

^c Depto. Ing. Electrónica UPV, Universidad Politécnica de Valencia, Espanha

como o de Householder ou o de Gram-Schmidt modificado [6], produz, a partir de uma matriz H , matrizes únicas Q e R tal que $H=QR$, onde R é uma matriz triangularizada. Uma vez determinadas estas matrizes, se pode trabalhar com o vetor de valores recebidos em Rx_1 e Rx_2 , r , e desta forma, estimar o vetor s de símbolos que foram transmitidos pelas antenas Tx_1 e Tx_2 . A Fig.2 a seguir esquematiza o processo de obtenção de s a partir de H e r :

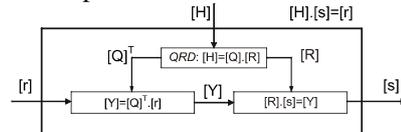


Fig.2: possuindo-se o canal estimado H e o vetor r recebido, é possível estimar-se os valores dos símbolos OFDM transmitidos, s .

III. CORDIC

Um método comumente utilizado para realizar as rotações de Givens se baseia no algoritmo iterativo CORDIC (COordinate Rotation Digital Computer) [7], que permite o cálculo de funções algébricas e trigonométricas de uma forma simples, rápida e precisa, realizando unicamente, de maneira iterativa, operações de soma, subtração e deslocamento de bits, evitando assim o uso de multiplicadores. Por esta razão é amplamente utilizado em VLSI para projetos de sistemas de processamento digital de sinais [8]. O algoritmo também ocupa uma área bastante reduzida em uma FPGA [9], fato que favoreceu o seu emprego no decodificador. Além disso, é versátil, pois é utilizado tanto na QRD quanto na divisão necessária na etapa seguinte do decodificador. A Fig.3 adiante explicita tais etapas.

IV. MODELO DE PONTO FIXO EM MATLAB

Se realizou em MATLAB um estudo do impacto causado pela quantificação de cada variável utilizada no algoritmo. Como exemplo, se toma a variável θ indicada na Fig.3, cuja quantificação – definida pelo *formato numérico t.f*, onde t é o número total de bits e f a parte fracionária – altera significativamente a resposta s do sistema (Fig.4). Cada variável quantificada aporta ao sistema um aumento na taxa de erros, como mostra a TABELA I.

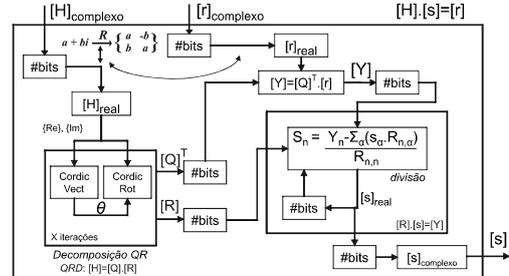


Fig.3: decodificador MIMO-OFDM. As etapas de decomposição QR e de divisão utilizarão CORDIC em sua implementação.

¹ No caso apresentado, Zero Forcing (ZF) e LS são equivalentes.

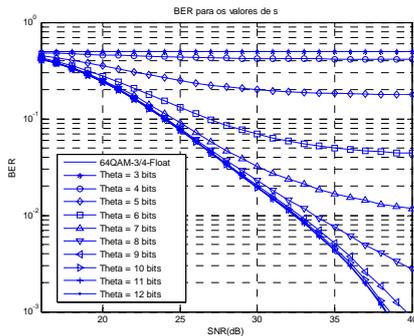


Fig.4: efeito da quantificação de θ : com 11bits fracionários, por exemplo, em $BER=10^{-3}$ se necessita uma SNR com 0,08dB a mais que para θ em ponto flutuante.

TABELA I: Perdas de implementação

Variável	Formato numérico	Perdas (dB) @ BER=10 ⁻³
H	12.8	0.16
r	12.8	0.09
θ	13.12	0.08
Q	11.10	0.10
R	13.9	0.10
Y	13.9	0.17
s(ext)	12.8	0.04

As perdas, porém, não são acumulativas, e o resultado final pode ser observado na próxima figura. Aqui se simulou a resposta do sistema para duas quantificações distintas:

- Quantificação 2: a definida na TABELA I;
- Quantificação 1: com um bit a menos de precisão que na quantificação 2 em todas as variáveis.

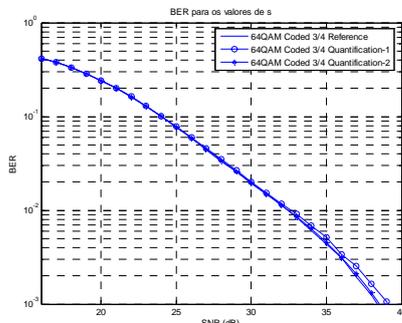


Fig.5: resposta s quando se variam as quantidades de bits das variáveis do sistema. Para a quantificação 1, há perdas de 0,75dB a $BER=10^{-3}$; já para a quantificação 2, as perdas são de somente 0,2dB nessa mesma taxa de erro.

V. IMPLEMENTAÇÃO EM HANDEL-C

Após a definição dos formatos numéricos de cada variável, se implementou o decodificador MIMO-OFDM usando a linguagem de programação de alto nível HANDEL-C. Esta permite gerar arquivos de implementação (*.bit através de *.vhd ou *.edf) em FPGAs e co-simulações com ACTIVE-HDL e MODELSIM, reduzindo assim o tempo gasto entre a programação e a implementação do sistema. Também permite, em alto nível, paralelizar processos dentro de um algoritmo, aumentando a velocidade do mesmo. A implementação em FPGA (Virtex2-5 8000) de cada código fornece os dados da TABELA II. Os resultados obtidos com a programação do código em HANDEL-C são os mesmos da simulação quantificada em MATLAB, mostradas na Fig.5.

TABELA II: Consumo hardware e velocidades

Código (*.edf)	Flip-flops	LUTs	Slices	Clock	Throughput
CORDIC	1.692	6.538	3.376	46,0MHz	46,0MHz
Decomp. QR	5.667	21.405	11.561	41,3MHz	234,7kHz
Dec. MIMO-OFDM	17.646	70.619	39.674	38,5MHz	218,8kHz

Internamente, há desvios menores ou iguais a duas vezes a precisão de cada variável; entretanto, estes não afetam o resultado final, como exemplifica a TABELA III:

TABELA III: Comparação de valores pontuais

Ambiente	MATLAB	DK Celoxica
Canal H	-733 + 558i -122 - 116i 823 + 281i	-733 + 558i -62 + 171i -122 - 116i 823 + 281i
Sinal recebido r	-665 - 347i -242 - 137i	-665 - 347i -242 - 137i
Matriz Q (QR)	800 609 -38 -182 -612 800 179 -38 133 -129 1006 -41 127 132 39 1004	799 609 -38 -181 -612 800 179 -38 133 -128 1006 -41 127 133 39 1006
Matriz R (QR)	-3748 -1 -162 4 0 -3748 -7 -162 -3 -2 3369 -5 0 -4 1 3367	-3747 -1 -160 5 -1 -3747 -6 -161 -2 0 3367 -3 0 -2 0 3368
Sinal enviado s (estimado)	102 + 178i -85 + 1i	102 + 178i -85 + 1i
Sinal enviado s (real)	102 + 179i -85 + 1i	(é o que se deseja encontrar)

Obs: os valores foram multiplicados pelos devidos fatores para uma melhor visualização da tabela.

VI. CONCLUSÕES

A metodologia empregada ao estudo de ponto fixo, assim como a implementação do decodificador se mostrou bastante precisa, gerando resultados satisfatórios. No entanto, o consumo hardware foi bastante elevado, e a velocidade proposta ainda não foi alcançada. Visando arquiteturas eficientes e a melhoria destes pontos, se sugere a aplicação de arrays sistólicos de processadores CORDIC para a QRD [10], bem como a utilização plena do paralelismo que oferece o HANDEL-C. Outra opção seria baixar o nível de abstração e trabalhar com linguagens estruturais (VHDL, por exemplo).

Finalmente, para futuros trabalhos, se propõe a generalização do decodificador para sistemas $M \times N$. O código implementado pode ser reutilizado também para detecção de MIMO-OFDM por Sphere Decoding, e com poucas mudanças, pode-se implementar um receptor Minimum Mean Square Error (MMSE) em vez do ZF.

REFERÊNCIAS

- [1] Technical Specification: Broadband Radio Access Networks (BRAN); HIPERLAN Type 2; Physical (PHY) layer". ETSI TS pp.101-475, 2001.
- [2] WLAN Medium Access Control and Physical Layer specifications: High Speed Physical Layer in the 5 GHz band, IEEE Std. 802.11, 1999.
- [3] Broadband MIMO-OFDM Wireless Communications; G.L. Stuber et al; PROCEEDINGS OF THE IEEE, VOL. 92, NO. 2, February 2004.
- [4] Performance Enhancements in OFDM-WLAN Systems using MIMO Access Techniques, E.R. de Lima, S.J. Flores, V. Almenar, A. Girona, 1st ISWCS, Mauritius, 2004.
- [5] Numerical Linear Algebra and Applications, Biswa Nath Datta, Brooks/Cole Publishing Company, 1995.
- [6] Matrix Computations; G. H. Golub and C. F. Van Loan; John Hopkins University Press, 2nd Ed., 1989.
- [7] CORDIC-Based VLSI Architectures for Digital Signal Processing, YU HEN HU, IEEE Signal Processing Magazine, July 1992
- [8] Xilinx: CORDIC V 3.0 LogicCore Specification. DS-249, May 2004.
- [9] Implementación eficiente en FPGA del algoritmo CORDIC para coordenadas circulares y lineales; F. Angarita, A Perez-Pascual, T. Sansaloni, J. Valls, FPL, 2005.
- [10] Design of Systolic arrays for QR Decomposition; M. Misra., R. Moona, International Conference on Computer System and Education, IISc, 1994.