International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 9 (2019) pp. 2143-2147
© Research India Publications. http://www.ripublication.com

VLSI Architecture of a Kaiser Bessel Window Function for MIMO system

M.Bavethra[1]

Dr. T. Perarasi[2]
Associate Professor

Department of Electronics & communication Engineering
Karpaga Vinayaga College of Engineering & Technology, Madhurantakam

Abstract

The Orthogonal frequency division multiplexing (OFDM) plays an important role in MIMO system. It is used to increase high SER rate in worst cases also. To achieve this we are designing OFDM physical layer both transmitter and receiver part by using CORDIC algorithm for vector rotation and Kaiser Bessel Window function. OFDM is implemented using the IEEE standard 802.11a for High-speed Physical Layer in the 5 GHz Band.

The channel is modelled as multipath combined with Gaussian noise of unit variance and zero mean. In this project, for the ease of analysis 2 channels are considered. The equalization is done by assuming that the channel parameters are known. Adaptive equalizer is used. Convolutional encoder and Viterbi decorder are used for the encoding and decoding purpose respectively.

SNR (Signal to noise ratio) vs SER (Symbol error rate) analysis of the 16-QAM modulation scheme is performed in various conditions such as cordic and non-cordic. To implement it in OFDMA systems number of channels should be increased. Finally the performance analysis of the above schemes is carried out.

Keywords: OFDM, Kaiser Bessel window function, CORDIC Algorithm, SNR, SER.

I. INTRODUCTION

MIMO technology has attracted attention in wireless communications, because it offers significant increases in data throughput and link range without additional bandwidth or transmit power. It achieves this by higher spectral efficiency (more bits per second per hertz of bandwidth) and link reliability or diversity (reduced fading). Because of these properties, MIMO is a current theme of international wireless research.

In MIMO, we use N number of Transmitting antennas to sends data over multiple paths, thereby increasing the amount of information, the system carries and the data is received by multiple antennas and recombined properly by other algorithms to recover the data at the receiver. MIMO is an underlying technique for carrying data. It operates at the physical layer, below the protocols used to carry the data, so its channels can virtually work with any wireless transmission protocol. For example, MIMO can be used with the popular IEEE 802.11 (Wi-Fi) technology[2]. For these reasons, MIMO eventually will become the standard for carrying almost all wireless traffic.

MIMO is the only economical way to increase bandwidth, range and will become a core technology in wireless systems. Assessing the performance of these algorithms requires detailed understanding of multiple-input multiple-output (MIMO) channels as well as models that capture their complex spatial behaviour.

II. KEY DESCRIPTION

A. MIMO-OFDM

Multiple-input, multiple-output orthogonal frequency-division multiplexing (MIMO-OFDM) is the dominant air interface for 4G and 5G broadband in case of wireless communications. It combines multiple-input, multiple-output (MIMO) technology, which multiplies capacity by transmitting different signals over multiple antennas, and orthogonal frequency-division multiplexing (OFDM), which divides a radio channel into a large number of closely spaced subchannels to provide more reliable communications at high speeds. Research conducted during the mid-1990s showed that while MIMO can be used with other popular air interfaces such as time-division multiple access (TDMA) and code-division multiple access(CDMA), the combination of MIMO and OFDM is most practical at higher data rates.

MIMO-OFDM is the foundation for most advanced wireless local area network (wireless LAN) and mobile broadband network standards because it achieves the greatest spectral efficiency and, therefore, delivers the highest capacity and data throughput[1]. Greg Raleigh invented MIMO in 1996 when he showed that different data streams could be transmitted at the same time on the same frequency by taking advantage of the fact that signals transmitted through space bounce off objects (such as the ground) and take multiple paths to the receiver. That is, by using multiple antennas and precoding the data, different data streams could be sent over different paths. Raleigh suggested and later proved that the processing required by MIMO at higher speeds would be most manageable using OFDM modulation, because OFDM converts a high-speed data channel into a number of parallel lower-speed channels.

B. CORDIC IN FFT BLOCK

The Coordinate Rotation Digital Computer (CORDIC) is an arithmetic technique, which makes it possible to perform two dimensional rotations using simple hardware components. The algorithm can be used to evaluate elementary functions such as cosine, sine, arctangent, sin h, cos h, tan h, ln and exp. CORDIC algorithm appears in many applications because it uses only primitive operations like shifts and additions to implement more complex functions. CORDIC [1] algorithm is an alternative method to realize the butterfly operation without using any dedicated multiplier hardware. CORDIC algorithm is very versatile and hardware efficient since it requires only add and shift operations, making it suitable for the butterfly operations in FFT [2]. Instead of storing the actual twiddle factors in ROM, the CORDIC based FFT processor needs to store only the twiddle factor angles in ROM for the butterfly operation. Additionally, the CORDIC-based butterfly can be twice faster than the traditional multiplier-based butterfly in VLSI implementations.

C. KAISER BESSEL WINDOW FUNCTION

Windowing plays a key role in signal processing applications such as spectral analysis [1–4], where it refers to the process of modifying the input samples before utilizing them for computing discrete Fourier transform (DFT), such that undesired effects in spectral domain due to data truncation i.e., spectral leakage and picket fence effect [8] can be minimized. Thus, the selection of appropriate window function for desired applications is based on their spectral characteristics. Since windowing is used before the FFT [4, 8] and STFT [10, 11] for digital spectral analysis system in real time application, and plenty of literatures presenting flexible [12, 13] and high speed architectures [14, 15] for implementing fast Fourier transform (FFT) are available. But none of them, to the knowledge of the authors, explicitly described efficient implementation of the window functions to meet the specification of FFT in terms of speed and flexibility in new trend of real time applications, except only recently, attempt has been made to implement a set of popular window functions in hardware using CORDIC arithmetic units. Except popular window functions such as Hanning, Hamming and Blackman, Kaiser-Bessel window [8] is also an obvious choice for better spectral characteristics and this is widely used in signal processing [5], communication and biomedical signal processing . Thus it becomes essential to have high throughput hardware realization for Kaiser-Bessel window functions to match the speed of the FFT architecture in order to develop an efficacious real time spectral analysis system. Depending on the intended application, the system designer can choose from a handful of window length and time-bandwidth product to satisfy the tradeoff between various parameters [8] such as accuracy of amplitude and spectral purity.

This is needless to mention all real time applications which require variable length FFT processor e.g. the STFT (Short Term Fourier Transform) is generally computed using variable length FFT processor which demand the variable length (flexible or adaptive) windowing architecture for preprocessing of signal before FFT. In this case, flexible windowing architecture is an obvious choice over ROM based or on chip RAM based implementation to compute windowing coefficients on the fly. To the knowledge of authors, there is no flexible VLSI/ASIC architecture for computing Kaiser-Bessel windowing coefficients available in the literature. In this context, authors in this paper have proposed CORDIC based flexible VLSI architecture for Kaiser-Bessel windowing to compute the coefficients on the fly and to meet the real time specifications in terms of throughput with little compromising for power and area compared to ROM based implementation.

III. PROJECT WORK DESCRIPTION

Figure 3.1 shows the general block diagram of OFDMA systems. The input is provided to the serial to parallel register. This serial to parallel register divides the input signal into many parallel outputs. The received parallel outputs used as an input for Inverse Fourier transform device. This device gives the transformed output to the parallel to serial register. This register converts back the parallel data into the serial one. Then the serial signal passes through the communication channel. Noises may be added in the signal while passing through the channel.

At the receiver end it is received by again the serial to parallel register the parallel output is then passed through a Fourier transform device which converts it into the original signal. But some amount of noise is also added in the signal. The FFT length used is 128bits. Out of which 112bits are the data and remaining 16 bits are zero paddings. This parallel output signal then passes through parallel to serial register so that a single output can be received.

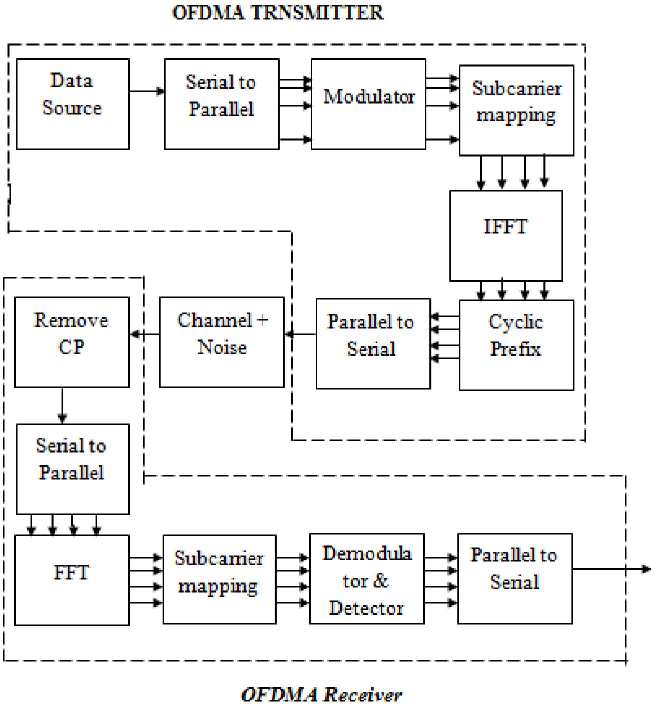


Figure 3.1: OFDMA block diagram

A. Generation of OFDM

To generate OFDM well, relationship between all carriers must be controlled carefully in order to control the orthogonality of carriers. For this motive, OFDM is generated by 1st choosing the spectrum required, depends on data input, and scheme of modulation used. For transmission, data is assign to every produced carrier. The required phase and amplitude of carrier is calculated and with the help of IFT converted back to its time domain. However in many applications, IFFT is used. IFFT performs the transformation efficiently.

B. Subcarrier modulation mappping

The OFDM subcarriers shall be modulated by using BPSK, QPSK, 16-QAM, or 64-QAM modulation, depending on the RATE requested. The encoded and interleaved binary serial input data shall be divided into groups of N_{BPSC} (1, 2, 4, or 6) bits and converted into complex numbers representing BPSK, QPSK, 16-QAM, or 64-QAM constellation points. In this paper we consider only 16-QAM as a sample for implementing CORDIC and KAISER-BESSEL functions. The conversion shall be performed according to Gray-coded constellation mappings with the input bit, b0, being the earliest in the stream. The output values, d, are formed by multiplying the resulting (I+jQ) value by a normalization factor K_{MOD}, as described in Equation (3.1).

$$d=\left(I+jQ\right)\times K_{MOD} \quad \text{--------------------------------}(3.1)$$

The normalization factor, K_{MOD}, depends on the base modulation mode, as prescribed in Table 3.1. Note that the modulation type can be different from the start to the end of the transmission, as the signal changes from SIGNAL to DATA, as shown in the frame format. The purpose of the normalization factor is to achieve the same average power for all mappings. In practical implementations, an approximate value of the normalization factor can be used, as long as the device conforms with the modulation accuracy requirements.

Table 3.1: Modulation-dependent normalization factor K_{MOD}

Modulation	**K_{MOD}**
BPSK	1
QPSK	1/√2
16-QAM	1/√10

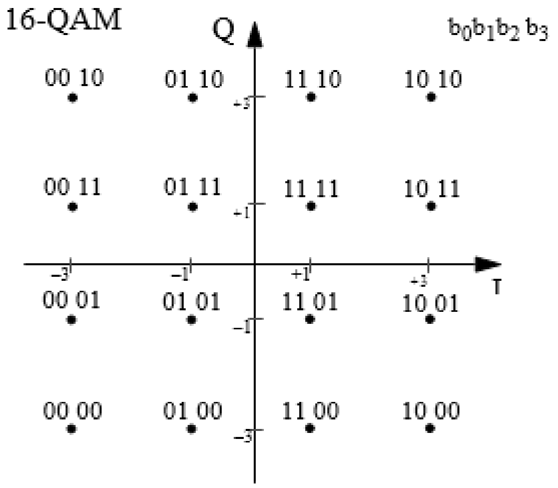


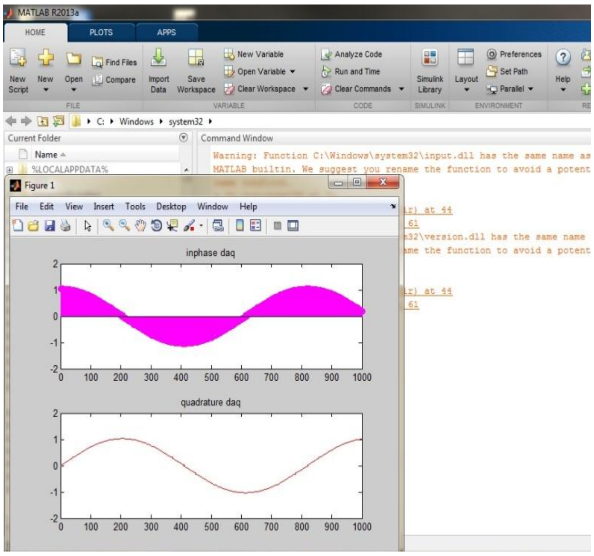
Figure 3.4: 16-QAM constellation bit encoding

For 16-QAM, b0b1 determines the I-value and b2b3 determines the Q value, as illustrated in Table 3.2

Table 3.2: 16-QAM encoding table

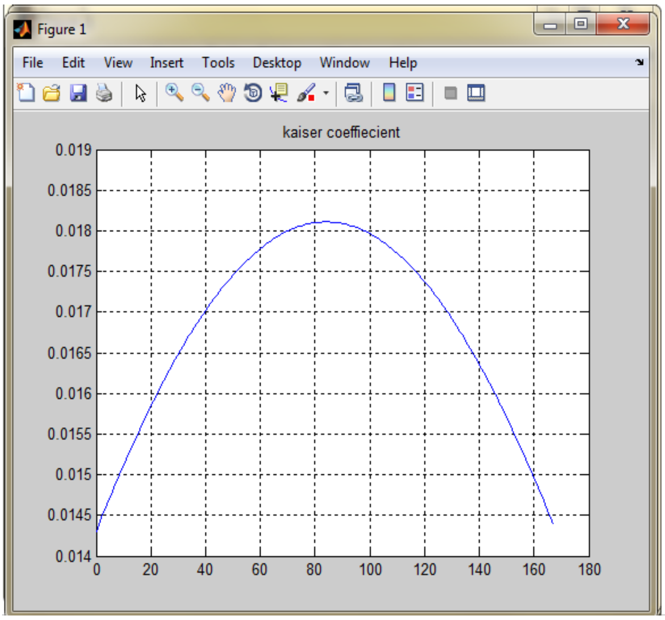
Input bits (b_0 b_1)	**I-out**	**Input bits (b_2)**	**Q-out**
00	–3	00	–3
01	–1	01	–1
11	1	11	1
10	3	10	3

IV. RESULTS AND DISCUSIONS

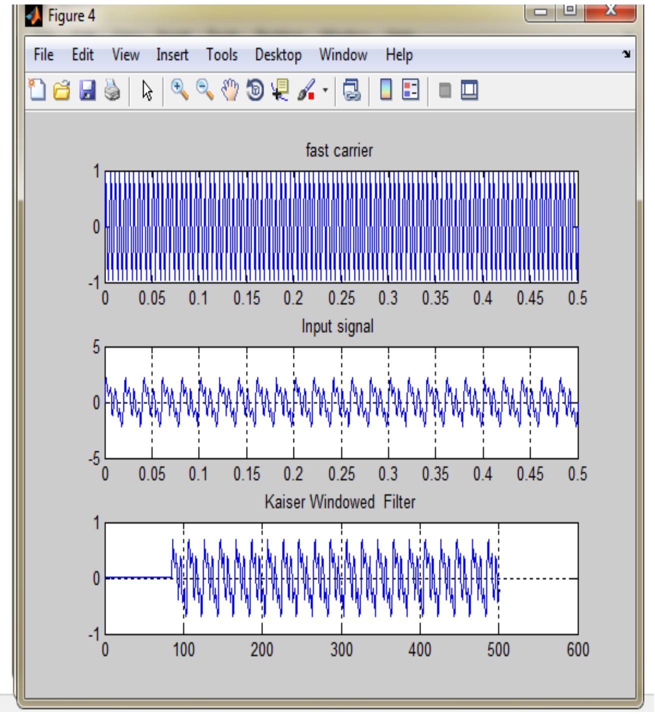


```
Warning: Function C:\Windows\system32\input.dll has the same name as
MATLAB builtin. We suggest you rename the function to avoid a potent
r) at 44
61
m32\version.dll has the same name
ame the function to avoid a potent
r) at 44
61
```

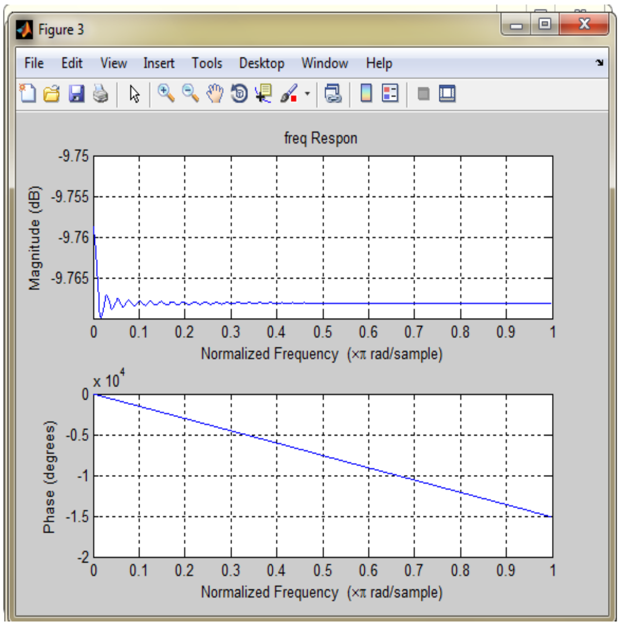
This waveform describes the inphase and quadrature of data aquisation syatem for vector rotation using the CORDIC algorithm.



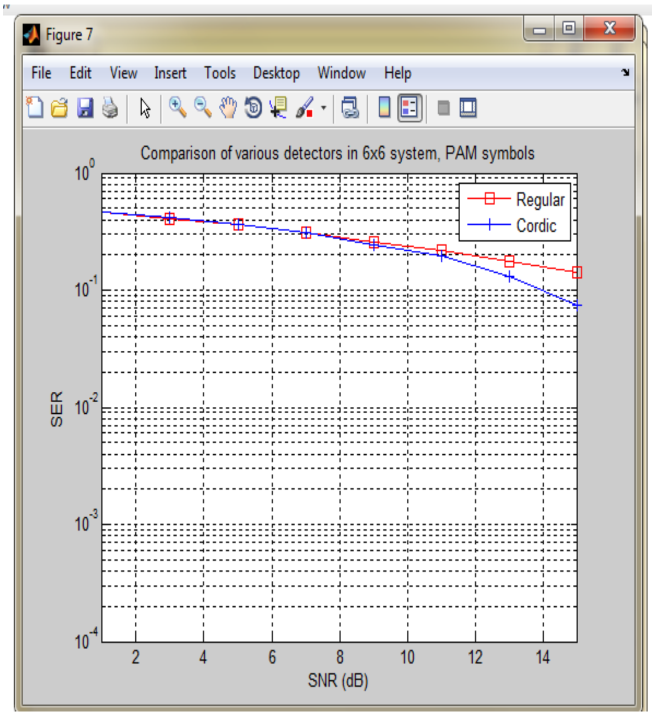
It is the waveform of Kaiser Co-efficient on behalf of the Bessel function implemented for the OFDM system which is plotted against the varying frequency



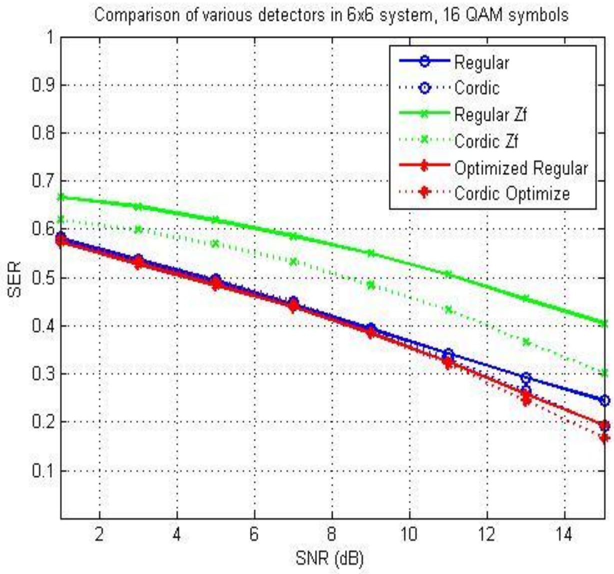
This shows the input message signal is carried by the fast carrier signal which undergoes the filtration process by using kaiser windowed filter then the filtered output occurred.



The frequency response of the modulation can be done by plotting the magnitude and phase of the signal against the normalized frequency.



SNR vs SER analysis of the modulation scheme 8-QAM as a sample in both regular and CORDIC conditions which shows the comparison between them.



SNR vs SER analysis of 16-QAM modulation scheme with the implementation of Bessel function was performed in various cases such as regular, Cordic, Optimized regular and optimized Cordic.

V. CONCLUSION

The OFDM is implemented using IEEE 802.11a with Kaiser-Bessel window function and Cordic algorithm and the SNR vs SER graph of 16-QAM was plotted with the above consideration and the performance was analysis as a better one compared to regular QAM modulation. In future this can be used in all the modulation schemes for the better performance.

As emerging technologies in wireless communication uses OFDM systems, this project can be used to achieve high data rates in the SNR environment. Future scope of this work is to implement spectrum sensing technique to identify strength signals among multipath data obtained at the receiver.

REFERENCES

[1] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, “Scaling up MIMO: Opportunities and challenges with very large arrays,” IEEE Signal Process. Mag., vol. 30, no. 1, pp. 40–60, Jan. 2013.

[2] S. Yang and L. Hanzo, “Fifty years of MIMO detection: The road to large-scale MIMOs,” Communications Surveys & Tutorials, IEEE, vol. 17, no. 4, pp. 1941–1988, Sep. 2015.

[3] W. Van Etten, “Maximum Likelihood Receiver for Multiple Channel Transmission Systems,” IEEE Trans. Commun., vol. 24, no. 2, pp. 276– 283, Feb. 1976.

[4] M. Kisialiou, X. Luo, and Z. Q. Luo, “Efficient Implementation of Quasi-Maximum-Likelihood Detection Based on Semidefinite Relaxation,” IEEE Trans. on Signal Process., vol. 57, no. 12, pp. 4811–4822, Dec. 2009.

[5] Y. Jiang, M. K. Varanasi, and J. Li, “Performance analysis of ZF and MMSE equalizers for MIMO systems: an in-depth study of the high SNR regime,” IEEE Trans. Inf. Theory, vol. 57, no. 4, pp. 2008–2026, Apr. 2011.

[6] C. Studer, S. Fateh, and D. Seethaler, “ASIC implementation of soft input soft-output MIMO detection using MMSE parallel interference cancellation,” IEEE J. Solid-State Circuits, vol. 46, no. 7, pp. 1754– 1765, Jul. 2011.

[7] L. Dai, X. Gao, X. Su, S. Han, Z. Wang et al., “Low-Complexity Soft- Output Signal Detection Based on Gauss-Seidel Method for Uplink Multi-User Large-Scale MIMO Systems,” IEEE Trans. Veh. Technol., vol. 64, no. 10, pp. 4839–4845, Oct. 2015.

[8] X. Gao, L. Dai, Y. Hu, Z. Wang, and Z. Wang, “Matrix inversion-less signal detection using SOR method for uplink large-scale MIMO systems,” in Proc. IEEE Global Telecommun. Conf. (IEEE GLOBECOM), Austin, TX, USA, Dec. 2014, pp. 3291–3295.

[9] M. Wu, B. Yin, G. Wang, C. Dick, J. R. Cavallaro, and C. Studer, “Large-scale MIMO detection for 3GPP LTE: algorithms and FPGA implementations,” IEEE J. Sel. Topics Signal Process., vol. 8, no. 5, pp. 916–929, Oct. 2014.

[10] G. Peng, L. Liu, S. Zhou, S. Yin, and S. Wei, “A 1.58 Gbps/W 0.40 Gbps/mm2 ASIC Implementation of MMSE Detection for 128_8 64- QAM Massive MIMO in 65 nm CMOS,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. PP, no. 99, pp. 1–14, 2017.

[11] G. Peng, L. Liu, P. Zhang, S. Yin, and S. Wei, “Low-Computing-Load, High-Parallelism Detection Method Based on Chebyshev Iteration for Massive MIMO Systems With VLSI Architecture,” IEEE Trans. Signal Process., vol. 65, no. 14, pp. 3775–3788, Jul. 2017.

[12] H. Prabhu, J. N. Rodrigues, L. Liu, and O. Edfors, “A 60pJ/b 300Mb/s 128_8 Massive MIMO precoder-detector in 28nm FD-SOI,” in Proc. IEEE Int. Solid-State Circuits Conf. Dig., San Francisco, CA, Feb. 2017, pp. 60–61.

[13] W. Tang, C.-H. Chen, and Z. Zhang, “A 0.58 mm2 2.76 Gb/s 79.8 pJ/b 256-QAM massive MIMO message-passing detector,” in Proc. IEEE Symp. VLSI Circuits, Honolulu, USA, Jun. 2016, pp. 1–2.

[14] X. Chu and J. Mcallister, “Software-Defined Sphere Decoding for FPGA-Based MIMO Detection,” IEEE Trans. Signal Process., vol. 60, no. 11, pp. 6017–6026, Nov. 2012.

[15] M. M. Mansour, S. P. Alex, and L. M. A. Jalloul, “Reduced Complexity Soft-Output MIMO Sphere Detectors-Part I: Algorithmic Optimizations,” IEEE Trans. on Signal Process., vol. 62, no. 21, pp. 5505–5520, Nov. 2014.