FPGA REALIZATION OF TIME-FREQUENCY DISTRIBUTION WITH COMPLEX-LAG ARGUMENT

Nikola Žarić, Irena Orović, Milena Zogović, Srdjan Stanković

Faculty of Electrical Engineering, University of Montenegro – Podgorica, Montenegro
email: zaric@cg.ac.yu

***Abstract* – Due to its ability to provide good concentration in the time-frequency plane, the fourth order complex-lag time-frequency distribution has been used for the analysis of signals with highly non-stationary instantaneous frequency. In order to make it available for practical applications, an architecture for realization of this distribution is proposed. The FPGA implementation is provided as well. Apart from the real-time processing, the proposed hardware realization is used to solve the problem of calculation errors that may appear due to the limited software precision.**

***Keywords* – time-frequency distributions, hardware implementation, FPGA**

1. INTRODUCTION

Time-frequency distributions have been used in different signal processing applications that deal with nonstationary signals. In order to provide good concentration along the instantaneous frequency various time-frequency distributions have been proposed [1]-[3]. An ideal concentration for linear frequency modulated signals could be achieved by using quadratic time-frequency distributions [4]-[6]. In the case of signals with highly non-stationary phase function, the quadratic distributions cannot provide satisfactory concentration. For this purpose, higher order time-frequency distributions should be used. In order to provide good concentration for signals with significant phase variations within a few samples, the complex-lag time-frequency distributions have been introduced [7]-[9]. However, the complex-lag distributions are computationally very complex and it would be difficult to provide a software realization with real-time processing. Also, limited software precision may cause errors during the calculation of signal with complex-lag argument.

In this paper we propose an efficient hardware implementation of the fourth order complex-lag time-frequency distribution. The proposed realization provides real-time processing and it is suitable for VLSI implementation. In order to provide high processing speed, the fixed-point and the floating-point arithmetic are combined. Additionally, the proposed system is designed to solve the problem of calculation errors that may appear in the software realization.

The proposed system consists of the circuits that are used for the calculation of: 1) complex-lag function that is used within the complex-lag distribution, 2) the product of real-lag signal terms and complex-lag function, and 3) the Fourier transform of this product. Due to its complexity, the main efforts are made to provide hardware realization of signal terms with complex-lag argument within the first step. In order to verify the results, the proposed system is implemented on the FPGAs chip.

2. THEORETICAL BACKGROUND

The fourth order complex-lag time-frequency distributions (CTD) has been defined as [7]:

$$CTD(t,w)=\int_{-\infty}^{\infty} x(t+\frac{\tau}{4})x^{*}(t-\frac{\tau}{4})x^{-j}(t+j\frac{\tau}{4})x^{j}(t-j\frac{\tau}{4})e^{-j\omega\tau}d\tau.$$

The spreading factor $Q(t,\tau)$ for $CTD(t,\omega)$ is [7]:

$$Q(t,\tau)=\phi^{(5)}(t)\frac{\tau^5}{4^4 5!}+\phi^{(9)}(t)\frac{\tau^9}{4^8 9!}+... \qquad (1)$$

The ideal concentration would be achieved for $Q(t,\tau)=0$. Note that, the dominant term in the spread factor is of the fifth order, which provides significant improvement compared to the Wigner distribution and L-Wigner distribution (Table I) [7]. Thus, for example, the CTD provides an ideal concentration along the IF for polynomial signals whose phase is up to fourth order. Additionally, for signals with higher order phase function, the fifth phase derivative in the case of the CTD is significantly reduced compared to the fourth order polynomial Wigner-Ville distribution (PWVD) (Table 1). Also, the next term in the spread factor for the CTD is of ninth order, while for PWVD this term is of seventh order (Table 1).

Table 1. Spread factor Q(t,τ) in some time-frequency distributions

Distribution	Spread factor
Wigner distribution	$Q(t,\tau)=\phi^{(3)}(t)\frac{\tau^3}{2^2 3!}+\phi^{(5)}(t)\frac{\tau^5}{2^4 5!}+\dots$
L-Wigner distribution	$Q(t,\tau)=\phi^{(3)}(t)\frac{\tau^3}{L^2 2^2 3!}+\phi^{(5)}(t)\frac{\tau^5}{L^4 2^4 5!}+\dots$
IV order PWVD	$Q(t,\tau)=-0.327\phi^{(5)}(t)\frac{\tau^5}{5!}-0.386\phi^{(7)}(t)\frac{\tau^7}{7!}+\dots$
CTD	$Q(t,\tau)=\phi^{(5)}(t)\frac{\tau^5}{4^4 5!}+\phi^{(9)}(t)\frac{\tau^9}{4^8 9!}+\dots$

In the discrete form CTD could be written as [7]:

$$CTD(n,k)=\sum_{m=-N/2}^{N/2-1} w(m)x(n+m)x^*(n-m)\times \\ \times x^{-j}(n+jm)x^{j}(n-jm)e^{-\frac{2\pi}{N}4mk}. \quad (2)$$

where $w(m)$ is a lag window of length N. In practical realization, the values of signal with complex-lag argument: $x(n+jm)$ and $x(n-jm)$, are not available. These values could be obtained by using the analytical extension of the signal with real valued argument. The analytical extension of signal $x(n)$ could be obtained as [7]:

$$x(n\pm jm)=\frac{1}{N}\sum_{k=-N/2}^{N/2-1}X(k)e^{j\frac{2\pi}{N}(n\pm jm)k}= \\ =\frac{1}{N}\sum_{k=-N/2}^{N/2-1}X(k)e^{\mp\frac{2\pi}{N}mk}e^{j\frac{2\pi}{N}nk}=IFFT\{X(k)e^{\mp\frac{2\pi}{N}mk}\}, \quad (3)$$

where the region of convergence is $|n+jm|<\infty$ for a finite N. In practical realization, one should carefully set up the necessary precision, since the term $e^{-\frac{2\pi}{N}mk}$ could obtain large values that are out of precision range [8].

The calculation of complex-lag function $c(n,m)=x^{-j}(n+jm)x^{j}(n-jm)$ within the CTD can be done according to [8]:

$$c(n,m)=e^{j\ln\left|\frac{x(n-jm)}{x(n+jm)}\right|}, \quad (4)$$

where the constant amplitude of $x(n)$ is assumed. Thus, (2) can be written in the form:

$$CTD(n,k)=FT_m\{w(m)x(n+m)x^{-1}(n-m)c(n,m)\}. \quad (5)$$

which will be used in the proposed hardware realization.

3. ARCHITECTURE AND FPGA REALIZATION

An architecture and FPGA realization of the CTD are proposed in this Section. The realization of the distribution is done according to (5), while its block scheme is shown in Fig 1. The samples of signal with real argument ($x(n+m)$ and $x(n-m)$) are fed to the input of the system. These samples are used to calculate the values of signal with complex-lag argument within the BLOCK 1. The BLOCK 2 is used to calculate the complex-lag function $c(n,m)$: it performs the calculation of the j-th power ($j=\sqrt{-1}$) of complex-lag signal terms. In parallel, the real-lag signal terms $x(n+m)$ and $x^*(n-m)$ are multiplied within the MULT circuit and then multiplied by the outputs of BLOCK 2. The outputs are fed to the input of FFT circuit and the complex-lag time-frequency distribution is obtained.

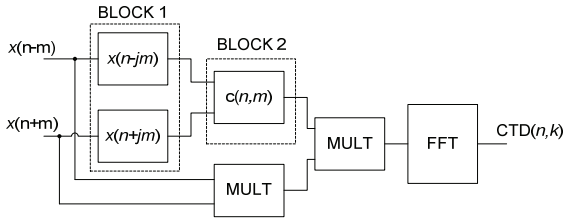


Fig. 1 – Block scheme for realization of CTD(n,k)

The implementation of the proposed architecture can be done on various hardware devices, such as digital signal processor, ASIC and FPGA chips. The FPGA implementation provide high speed processing, reconfigurability, low producing time and costs requirements, inbuilt special hardware such as RAM, etc. Thus, for implementation of the proposed system we have considered the FPGA chips. Particularly, we use the Quartus II v8.1 software and the EP2S15F484C3 chip (Stratix II family) from Altera [12].

3.1. Realization of BLOCK 1 (calculation of signal with complex-lag argument)

An architecture and the corresponding FPGA realization that are used to calculate the signal with complex-lag argument are presented in this subsection. The realization is done throughout three steps, as it is shown in Fig. 2. The first step is used to calculate the Fourier transform of signal with real argument $s(n\pm m)$. The product of Fourier transform samples $X(k)$ and $e^{-\frac{2\pi}{N}mk}$ coefficients is calculated within the STEP 2, while the STEP 3 represents the calculation of the inverse Fourier transform.

STEP 1: The signal samples are fed to the input of the FFT circuit. These samples are represented in the fixed-point format by using 16 bits, since it is common case in practical applications. At the output of FFT circuit, we obtain the Fourier transform ($X(k)$) of signal that is represented in the fixed-point format, as well.

The FFT circuit is realized by using the FFT MegaCore function that has been proposed by Altera [13]. For device EP2S15F484C3 (Stratix II family) and the 16-bit fixed point format, the FFT chip requires: 3467 ALUTs, 2534 memory bits, 24 DSP block 9-bit elements, and transform time 20*n*s. The

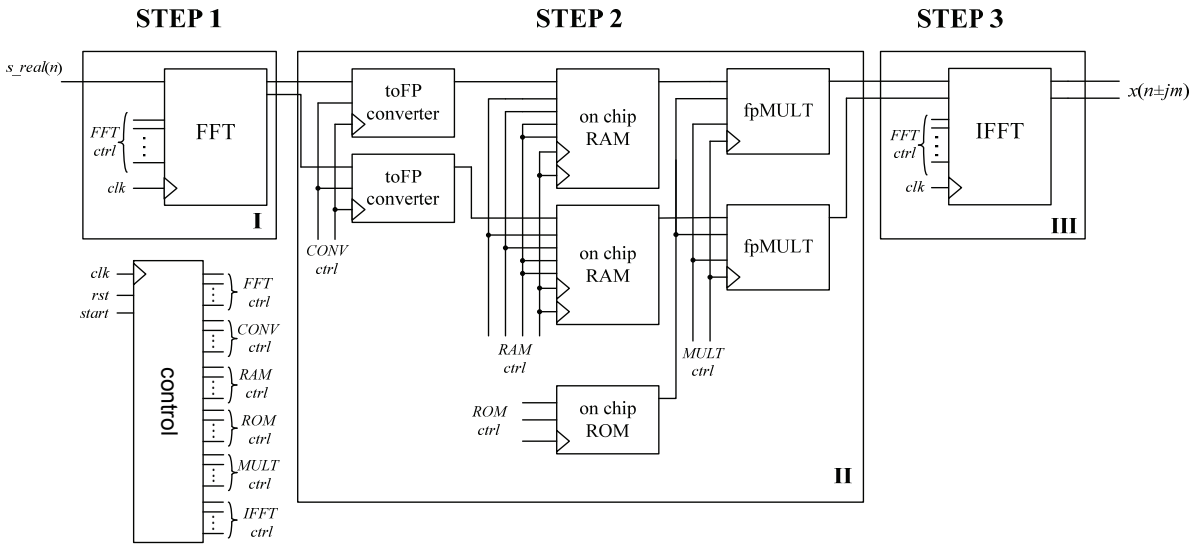
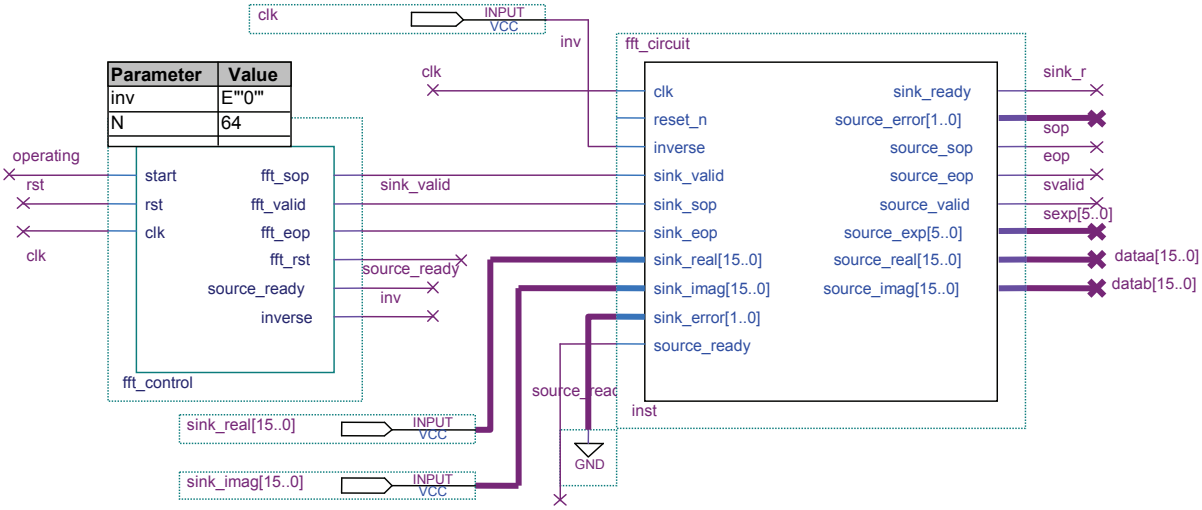
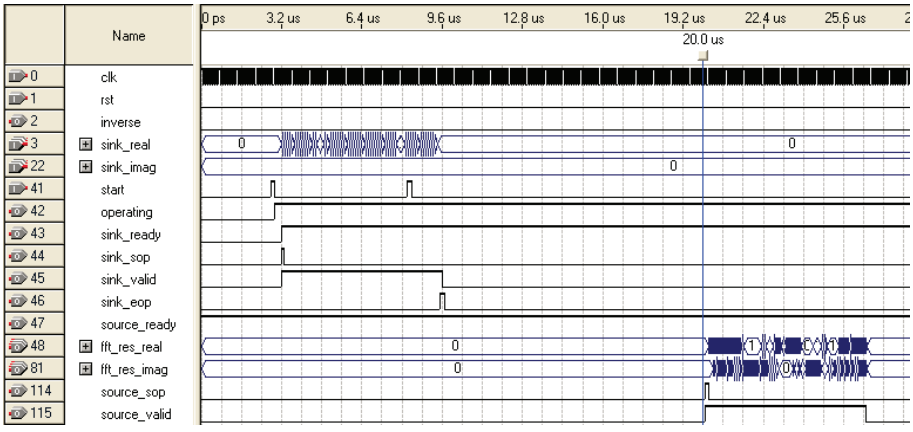


Fig. 2 – Architecture for calculation of signal with complex-lag argument



(a)



(b)

Fig. 3 – (a) FFT circuit and its control logic, (b) Simulation results for the FFT circuit

block symbol of the FFT circuit and its synchronization signals are shown in Fig. 3.a. Simulation results, including control signals are given in Fig. 3.b. The input signal is represented in fixed-point format with 16-bits, while the transform length is 64.

STEP 2: This step performs the multiplication of $X(k)$ and the corresponding coefficients $e^{-\frac{2\pi}{N}mk}$. However, the coefficients $e^{-\frac{2\pi}{N}mk}$ are represented in floating-point format, since they could obtain large values. Thus, $X(k)$ should be firstly converted to floating-

point format. In order to avoid miscalculation errors, the floating-point format is chosen to provide a satisfying precision even for large values of real exponential term $e^{-\frac{2\pi}{N}mk}$ i.e. for large m and k. Namely, the required precision depends on the number of input samples N, since $-N/2 \le m \le N/2-1$ and $-N/2 \le k \le N/2$ holds. For example, if N=64 the extended single precision IEEE-754 format should be used, while for N=128, the extended double precision IEEE-754 format is required.

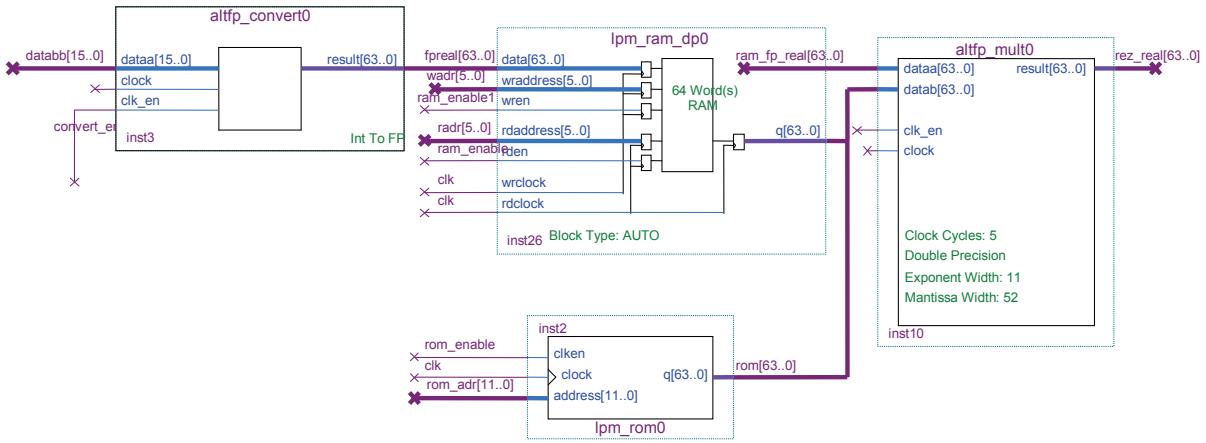
In order to adjust the format of $X(k)$ samples and $e^{-\frac{2\pi}{N}mk}$ coefficients, it is necessary to perform conversion of $X(k)$ samples from fixed-point to the floating-point format. The FP_converter circuit from Fig. 2, is used for that purpose. It is realized by using ALTFP_CONVERT circuit. The enable signal for this circuit is source_valid signal that is generated at the output of FFT circuit. For conversion from 16-bit fixed-point format to double floating-point format this circuit requires 37 ALUTs and 223 memory bits with the latency of six clock cycles.

After conversion $X(k)$ samples are stored on chip RAM memory (read access memory). Namely, the same $X(k)$ samples are used to calculate analytical extension for various values of parameter m. Thus, it is more appropriate to store $X(k)$ samples in memory than to calculate FFT for each m. The RAM memory is realized by using LMP_RAM circuit that requires 4096 memory bits.. The write_address and write_enable signal are controlled by the signals from FP_convereter circuit, while read_address and read_enable signals are the same as for ROM memory.

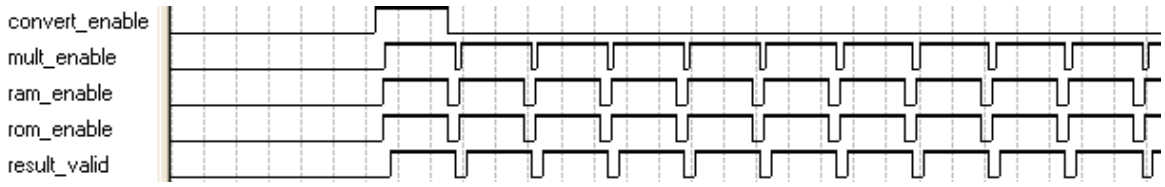
The coefficients $e^{-\frac{2\pi}{N}mk}$ are stored in the on chip read only memory (ROM). It is designed by using LMP_ROM device that requires 262144 memory bits for N=64.

The values $X(k)$ and $e^{-\frac{2\pi}{N}mk}$, obtained from RAM and ROM memories, respectively, are multiplied by using ALTFP_MULT circuit (floating point multiplier fpMULT in Fig. 2.). This circuit has the latency of five clock cycles, and it requires 235 ALUTs, 342 registers. The block symbols and time-line of control signals for FP_convert circuit, RAM and ROM memories, and floating-point multiplier are shown in Fig 4.

STEP 3: In order to obtain complex-lag signal, the inverse Fourier transform is calculated for the outputs of fpMULT circuits from the previous step. The IFFT circuit is obtained by using the FFT MegaCore function, similarly as the FFT circuit. The only difference is that IFFT circuit should operate with floating-point number format. For device EP2S15F484C3 (Stratix II family) it requires 13760 ALUTS, 4786 memory bits and 64 DSP block 9-bit elements.



(a)



(b)

Fig. 4 – a) Block symbols for circuits used in STEP 2, b) corresponding control signals

3.2. Realization of BLOCK 2 (calculation of the complex-lag function)

The realization of the complex-lag function is done according to (4). The corresponding block scheme is given in Fig 5.

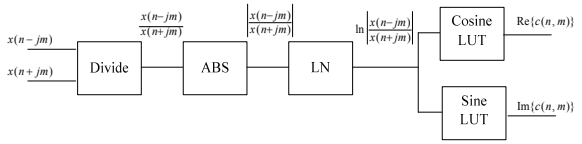


Fig. 5 Block scheme for realization of BLOCK 2

The term $x(n\text{-}jm)/x(\underline{n}+jm)$ is obtained by using ALTFP_DIV circuit, that requires 124 ALUTs, 4608 memory bits, and has the latency of six clock cycles. The enable signal for this circuit is source_valid signal that is generated at the output of IFFT circuit. Further, the ALTFP_ABS circuit (with the latency of 1 clock cycles) is used to calculate the absolute value of $x(n\text{-}jm)/x(\underline{n}+jm)$, and the output is fed to the input of LN circuit to calculate the natural logarithm. The ALTFP_LOG circuit has been recently proposed by Altera, and it is included in the Quartus v8.1. The latency of this circuit is 21 clock cycles and it requires 2132 ALUTs, 16 DSP blocks and 1948 registers. The exponential function is calculated bz using sine and cosine function as follows: $e^{jLN} = \cos(LN) + j\sin(LN)$, where $LN = \ln\left|\frac{x(n-jm)}{x(n+jm)}\right|$.

The values of cosine and sine functions are obtained by using look up table (LUT). In order to provide satisfying precision, the LUTs contain 512 values represented with 16 bits. Thus, at the output of BLOCK 2, we obtain the complex-lag function $c(n,m)$ represented by 16 bits in fixed-point format.

Finally, $c(n,m)$ is multiplied by $x(n+m)x(n-m)$ within the standard LMP_MULT circuit and fed to the input of the FFT circuit. The time-frequency distribution with complex-lag argument is obtained at the output of FFT circuit.

The available characteristic of EP2S15F484C3 chip and the resources utilized for implementation of the proposed system, are given in Table 2.

Table 2. Chip characteristics for the proposed system

EP2S15F484C3	**No. of pins**	**Logic elements**	**clk : f_{max}**	**Power cons.**
Available	343	12480	500 MHz	1.2 V
Utilized	194 (57%)	9856 (79%)	249 MHz	1.2 V

4. CONCLUSION

The FPGA realization of the fourth order complex-lag time-frequency distribution is proposed. It enables a real-time processing, and consequently, provides an opportunity to use good properties of this distribution in various applications. The fixed-point and floating-point formats are combined in order to provide a solution for high speed processing with satisfying precision. The chip parameters and simulation results are presented as well. The proposed realization can be further extended and used as a base for implementation of the complex-lag time-frequency distribution of an arbitrary order.

ACKNOWLEDGEMENT

This work is supported by the Montenegrin Ministry for Science and Education.

5. REFERENCES

[1] L. Cohen, “Time-frequency distributions – a review,” Proc. IEEE, vol. 77, no. 7, pp. 941-981, July 1989.

[2] Proceedings of the IEEE, Special issue on Time-Frequency Analysis, vol. 84, no. 9, Sept. 1996.

[3] B. Boashash, Time-frequency signal analysis, Elsevier, Oxford, UK, 2003.

[4] LJ. Stanković, “A method for improved energy concentration in the time-frequency signal analysis using the L-Wigner distribution,” *IEEE Trans. on Signal Processing*, vol. 43, no. 1, pp. 1262-1268, May 1995.

[5] B. Boashash and P. O’Shea, “Polynomial Wigner-Ville distributions and their relationship to time-varying higher order spectra,” *IEEE Trans. on Signal Processing*, vol. 42, no. 1, pp. 216-220, Jan. 1994.

[6] B. Barkat, “Instantaneous frequency estimation of non-linear frequency modulated signals in the presence of multiplicative and additive noise,” *IEEE Trans. on Signal Processing*, vol. 49, no. 10, pp. 2214 – 2222, Oct. 2001.

[7] S. Stankovic and LJ. Stankovic, “Introducing time-frequency distribution with a "complex-time" argument,” *Electronics Letters*, vol. 32, no. 14, pp. 1265-1267, July 1996.

[8] LJ. Stankovic, Time-frequency distributions with complex argument," *IEEE Trans. on Signal Processing*, vol. 50, no. 3, pp. 475-486, Mar. 2002.

[9] M. Morelande, B. Senadji, and B. Boashash, “Complex-lag polynomial Wigner-Ville distribution,” *Proc. of* IEEE *Speech and Image Technologies for Computing and Telecom*., vol. 1, pp.43-46, Dec. 1997.

[10] C. Cornu, S. Stanković, C. Ioana, A. Quinquis, and LJ. Stanković, “Generalized representation derivatives for regular signals,” *IEEE Trans. on Signal Processing*, vol. 55, no. 10, pp. 4831-4838, Oct. 2007.

[11] S. Stanković, N. Žarić, I. Orović, and C. Ioana, “General form of time-frequency distribution with complex-lag argument," *Electronics Letters*, vol. 44, no. 11, pp. 699-701, May 2008.

[12] Stratix II, device datasheet: DC and Switching Characteristic, *Altera Corporation*, 2007.

[13] FFT MegaCore function, *Altera Corporation*, May 2008