An architecture for the VLSI design of systems for time-frequency analysis and time-varying filtering

Srdjan Stanković, LJubiša Stanković, Veselin N. Ivanović, Radovan Stojanović

Abstract— **A flexible system for time-frequency signal analysis is presented. It is based on the S-method, which has a significant advantage in implementation since it can involve, as a key intermediate step, the Short-time Fourier transform or the Hartley transform, each widely studied and commonly used in practice. Signal invariant and signal dependent system forms are presented. Hardware design, for a fixed-point arithmetic, is well-structured and suitable for VLSI implementation. The same hardware, without additional time requirements, may be shared for the simultaneous realization of the fourth order L-Wigner distribution, as well as for the realization of the cross-terms free fourth order polynomial Wigner-Ville distribution. This possibility makes the designed hardware suitable for wide range of the applications. The proposed hardware is applied to the realization of time-varying filtering, as well. Finally, it has been implemented with FPGAs chips (Field Programable Gate Array) in order to verify the results on real devices.**

I. Introduction

The increasing demands for high-speed real-time applications of signal processing algorithms, including time-frequency analysis, require an efficient hardware implementation of developed algorithms. The short-time Fourier transform (STFT) is the simplest tool for time-frequency analysis, and has been conventionally used for that purpose. However, its serious drawback is low concentration in time-frequency plane, which may be inconvenient in many applications. Various quadratic time-frequency distributions (TFDs), the most important of which is the Wigner distribution (WD), have been used to improve the resolution [11], [15]. The WD gives better concentration than the STFT, but it generates cross-terms in the case of multicomponent signals, what limits its applicability. Based on the unified definition of two most important and commonly used tools in time-frequency signal analysis, the STFT and the WD, the S-method (SM) for time-frequency signal analysis is proposed, [29], and used in [8], [14], [27]. It reduces or completely eliminates cross-terms without degrading high resolution of the WD. The fact that this method uses, as an intermediate step, the STFT for which the hardware implementations already exist, makes it very attractive for implementation. For real-valued signals implementation, the Hartley transform (HT) can be more appropriate tool than STFT. The SM form based on the HT is derived, and it has been used for the hardware design.

In order to improve concentration of signals whose instantaneous frequency is highly nonlinear, higher order TFDs have been defined, [7], [26], [30]. Some of them, like the L-Wigner distribution (L-WD) and the polynomial Wigner-Ville distribution (PWVD), can be realized based on the SM.

All time-frequency representations, beyond the STFT, are numerically quite complex and require significant calculation time. This fact makes them unsuitable for real-time analysis, and severely restricts their application. Hardware implementations, when they are possible, can overcome this problem and enable application of time-frequency methods in numerous additional problems in practice.

In this paper, we develop an architecture suitable for the VLSI implementation of the SM and higher order TFDs. The discrete form of the SM reads [29], [32]:

$$SM(n,k) = \sum_{i=-L_d}^{L_d} P(i)$$

Annales des Télécommunications, vol.57, No.9/10, Septembre/Octobre 2002.

$$\times STFT(n,k+i)STFT^{*}(n,k-i), \qquad (1)$$

where STFT of the analyzed signal $f(n)$ is defined as:

$$STFT(n,k)=\sum_{i=-N/2+1}^{N/2} f(n+i)w(i)e^{-j\frac{2\pi}{N}ik}. \qquad (2)$$

Width of the frequency domain window $P(i)$ is $W_P = 2L_d + 1$. For the marginal cases of window width: *a)* $L_d = N/2$, the pseudo WD is obtained, *b)* $L_d = 0$, the spectrogram follows. Note that by an appropriate selection of the window $P(i)$ width, the same concentration as in the WD case can be achieved, avoiding the presence of cross-terms. Influence of the window $P(i)$ width to the TFD concentration is analyzed in [29], [32], [35]. The algorithms for a signal-independent and a signal-dependent realization of the SM are given in [35].

A complete hardware for the SM based algorithm, including the STFT and the HT, for the efficient ASIC implementation, is presented. As it will be shown, hardware design for realization of the higher order TFDs enables one to preserve the same processing time as in the SM case.

The paper is organized as follows. The hardware design for the SM realization (in the cases of its signal independent and signal dependent forms) based on the STFT is presented in Section II. In the Section II we partly represent a review of the results from [25], [35] with some new architectures. After that, hardware design for the corresponding SM realization based on the HT is proposed. In Section IV hardware for the higher order TFDs realization is designed. Real-time implementation of the time-varying filtering is proposed in Section V. The error analysis is performed in Section VI. The proposed approaches are verified in Section VII by designing the FPGA chips, based on parallel and serial computation architectures. For both cases the implementation results are given.

II. Hardware for the S-method Implementation Based on the STFT

A. Signal Independent Window, [25]

In this section, we present a system for the SM implementation assuming fixed window $P(i)$ width $W_P = 2L_d + 1$. A scheme of the proposed architecture, based on (1) and (2), is shown in Fig. 1a). It consists of two blocks. The first block is used for the STFT implementation, and the second block is used to modify the STFT in order to obtain the improved TFD concentration, based on the SM. The STFT can be implemented by using the available FFT chips or the approaches based on the recursive algorithms (both of them are widely known in the literature and practice). The design presented here is based on the recursive algorithm, [3], [19], [24], [29], which is, due to the reduced hardware complexity, more suitable for VLSI implementation [19]. Assuming a rectangular window $w(i)$, the STFT can be written as:

$$\begin{aligned} &STFT(n,k)\\ &=[f(n+N/2)-f(n-N/2)](-1)^{k}\\ &\quad +STFT(n-1,k)e^{j2\pi k/N}. \end{aligned} \qquad (3)$$

A complete system consists of N channels described by (3), with $k = 0, 1, 2, ..., N-1$. For the cases of Hanning or Hamming window $w(i)$, the STFT is obtained by modifying $STFT(n,k)$ in (3) as follows:

$$\begin{aligned} &STFT_H(n,k) = a_{-1}STFT(n,k-1)\\ &+a_0 STFT(n,k) + a_1 STFT(n,k+1) \end{aligned} \qquad (4)$$

where the coefficients a_{-1}, a_0, a_1 are: $(0.25, 0.5, 0.25)$ and $(0.23, 0.54, 0.23)$ for the Hanning and the Hamming window, respectively, [19], [29]. Each channel described by equation (3), involves complex multiplication and can be separated in two sub-channels involving only real computations. In order to describe these channels, we modify equation (3), using

$$STFT(n,k) = STFT_{\mathrm{Re}}(n,k) + jSTFT_{\mathrm{Im}}(n,k),$$

where $STFT_{\mathrm{Re}}(n,k)$ and $STFT_{\mathrm{Im}}(n,k)$ are the real and imaginary part in $STFT(n,k)$,

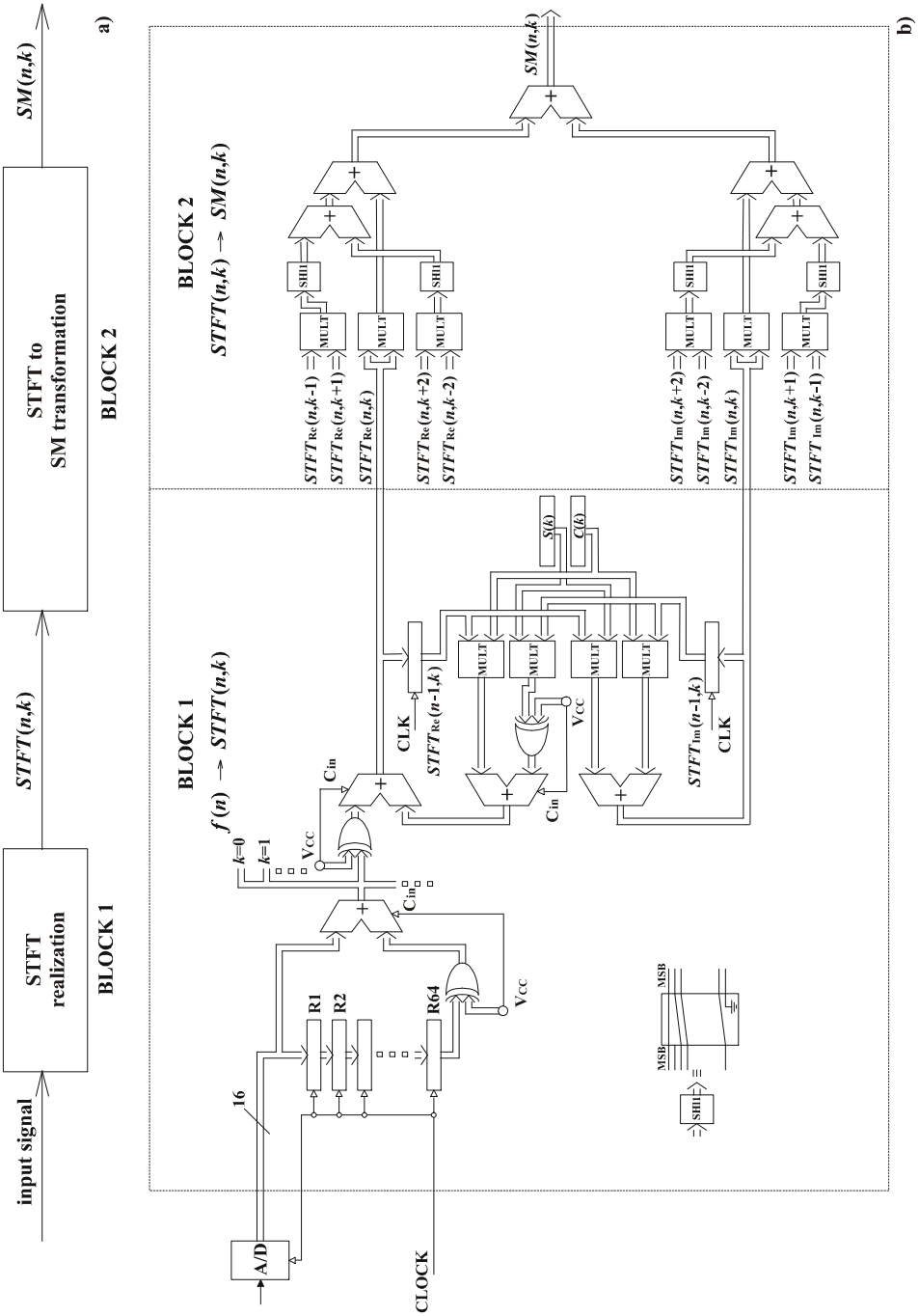

Fig. 1. Architecture for the signal independent S-method realization: a) Block scheme, b) Hardware (one channel with $L_d = 2$).

respectively. The equations describing these two sub-channels are then:

$$STFT_{\mathrm{Re}}(n,k) = F(n)(-1)^k + c(k)STFT_{\mathrm{Re}}(n-1,k) - s(k)STFT_{\mathrm{Im}}(n-1,k),$$
$$STFT_{\mathrm{Im}}(n,k) = c(k)STFT_{\mathrm{Im}}(n-1,k) + s(k)STFT_{\mathrm{Re}}(n-1,k) \quad (5)$$

where: $c(k) = \cos(2\pi k/N)$; $s(k) = \sin(2\pi k/N)$ and $F(n) = f(n+N/2) - f(n-N/2)$. Now (1) can be written as:

$$SM(n,k) = |STFT(n,k)|^2 + 2\sum_{i=1}^{L_d} STFT_{\mathrm{Re}}(n,k+i)STFT_{\mathrm{Re}}(n,k-i) + 2\sum_{i=1}^{L_d} STFT_{\mathrm{Im}}(n,k+i)STFT_{\mathrm{Im}}(n,k-i). \quad (6)$$

This equation is used to modify the outputs of the STFT block in order to obtain the SM based TFD.

The hardware necessary for one channel implementation and for $L_d = 2$ is presented in Fig. 1b). It has been designed for a 16 bit fixed-point arithmetic. The total number of multipliers is $2(L_d + 3)$ and the total number of adders is $2(L_d + 2)$. The multiplication operation results in two sign-bit and, assuming Q15 format (15 fractional bit), the product must be shifted left by one bit to obtain correct results. This shifter is included as a part of multiplier. The throughput of the system is N. The longest path is one that connects the register storing $STFT(n-1, k\pm L_d)$, through 2 multipliers and $L_d + 3$ adders, with the output $SM(n,k)$. This path determines the fastest sampling rate. It can be observed that the SM implementation introduces only an additional delay of L_d adders, as compared to the spectrogram implementation. Thus, the fastest sampling rate (with parallel architecture) is essentially the same for both implementations (spectrogram and SM). This design can be implemented as an ASIC chip to meet the speed and performance demands of very fast real-time applications. Note that the architecture presented in Fig. 1 is of parallel form, since this form directly follows from the equations. Its modification into a serial form is given in Section VII.

B. Signal Dependent Window

Fixed width of the window $P(i)$ has its disadvantages. Namely, in the analysis of multicomponent signals, having different widths of the STFT's components, selection of the fixed window width may be inappropriate for some of the components. Too short window would results in lower concentration, while too long window might have the negative effects regarding the cross-terms, [29], and noise influence, [32]. This motivates the introduction of a signal dependent window width, which will be, in discrete domain, denoted by $W_P(n,k) = 2L_d(n,k) + 1$, [35]. It should, for each n and k, follow the widths of the STFT of the signal components. The window width should include the summation in (1) over the terms where $STFT(n,k+i)$ and $STFT^*(n,k-i)$ are different from zero. But, the variable width of $P_{(n,k)}(i)$ should exclude the summation where one or both of the previous components are equal to zero, and, in addition, it should stop the summation outside a component. The variable window $P_{(n,k)}(i)$ width will be implemented by using logic circuits that will turn off all lines $i \geq i_0$ for a given k when any $STFT(n,k+i_0)$ or $STFT^*(n,k-i_0)$ are equal to zero, or its absolute value is less than or equal to an assumed reference value R (i.e., if $|STFT(n,k)| \leq R$) [35]. In practice the reference value is selected based on a simple analysis of the signal and the implementation system.

If the implementation system is based on a fixed-point arithmetic, the signal should be properly scaled in order to optimally utilize available registers, [16], [23]. In the STFT case, the signal should be scaled to keep the output of the system within the range $[-1, 1]$. This allows one to apriori establish the reference value as a fraction of the maximum allowable value of the STFT. In the process of selecting the reference, one should not be concerned about an exact value, since the results are not too sensitive with regard to the ref-

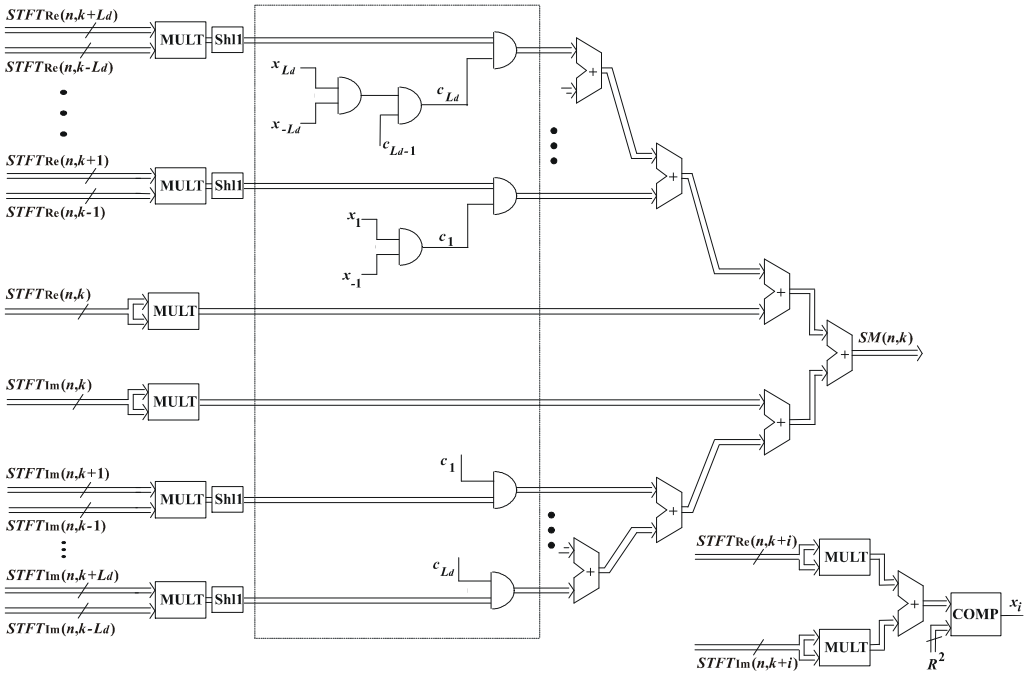

Fig. 2. Hardware modification for the signal dependent S-method realization.

erence value. For example, variation of the reference value between 10% and 25% of the STFT's maximum would hardly be noticed in the resulting time-frequency presentation. Thus, the reference value is a fraction of the maximal value of $|STFT(n,k)|$ for all k, i.e. $R = 1/Q$, where $1 \leq Q < \infty$. Generally, if we do not have apriori knowledge about the signal range, the reference value is defined as a fraction of the STFT's maximum, at a given instant t, $R = \max_k |STFT(n,k)| / Q$.

In order to accommodate for signal dependent window width, BLOCK 2 of the hardware in Fig. 1b) should be modified as shown in Fig. 2 (modification is emphasized with dashed lines). The quantities $x_{\pm i}$ $(i = 0, 1, ..., L_d)$ denote the outputs of the comparator with the inputs $|STFT(n, k \pm i)|^2$ and R^2 and have the value $x_i = 1$ if $|STFT(n, k+i)| > R$ and zero otherwise. The circuits needed to generate $x_{\pm i}$ are shown in Fig. 2. The control signals c_i that determine whether or not the $i-th$ term goes into summation in (6), are defined by

$$c_i = \prod_{m=1}^{i} x_{k+m} x_{k-m} = c_{i-1} x_{k+i} x_{k-i},$$

where we may assume (without loss of generality) that the maximal possible window $P(i)$ width is $2L_{d\max} + 1$ (variable window width approach with the predefined maximal window width). **The spectrogram value will be forwarded to the output, even if $|STFT(n,k)| \leq R$ for all k, so $c_0 \equiv 1$.**

In case of high noise, the value $x_i = 0$ ($|STFT(n,i)| \leq R$) may occur even if the non-noisy value $|STFT(n,k)|$ is greater than the reference value. We may just disregard this, being aware that the value of spectrogram is forwarded to the output anyway, or we may introduce a slightly more complicated logic function which will stop the summation in (6) only after two subsequent zeros in x_i and x_{i+1} are detected. In the latter case, control signals c_i are defined in the same way, with x_m now being replaced by $\chi_m = x_m + x_{m+1}$ (" + " denotes a logical OR operation). The last modification may be used not only for

noisy signals, but also in the case when the STFT may assume zero (or less than reference R) values within a single auto term.

The signal dependent architecture can provide better time-frequency concentration of the signal, while with respect to the computation it is not simpler, and even requires more logic elements that the signal independent one.

III. Hardware for the S-method Implementation Based on the HT

When a signal is real, then one may use the HT instead of the STFT. The HT gives the real coefficients of the real analyzed signal, as opposed to the STFT. A discrete form of the short-time HT is given by, [10], [17], [20], [21]:

$$HT(n,k) = \sum_{i=-N/2+1}^{N/2} f(n+i)w(i)(\cos\frac{2\pi}{N}ik + \sin\frac{2\pi}{N}ik). \quad (7)$$

Note that the relation between $STFT(n,k)$ and $HT(n,k)$ is:

$$HT(n,k) = H_p(n,k) + H_n(n,k) = STFT_{\mathrm{Re}}(n,k) + STFT_{\mathrm{Im}}(n,k), \quad (8)$$

where:

$$H_p(n,k) = [HT(n,k) + HT(n,-k)]/2,$$
$$H_n(n,k) = [HT(n,k) - HT(n,-k)]/2. \quad (9)$$

Now, a recursive relation (similar to (5)) for the HT and rectangular window $w(i)$ can be derived:

$$HT(n,k) = F(n)(-1)^k + c(k)HT(n-1,k) + s(k)HT(n-1,-k), \quad (10)$$

where $HT(n,-k) = HT(n,N-k)$. For any other than the rectangular window, the HT should be modified in exactly the same manner as the STFT, adding coefficients a_0, $a_{\pm 1}$, ... Based on (6) and (8)-(9), the SM can be written in terms of the HT,

$$SM(n,k) = \frac{1}{2}[HT^2(n,k) + HT^2(n,-k)] + \sum_{i=1}^{L_d} P(i)[HT(n,k+i)HT(n,k-i) + HT(n,-k+i)HT(n,-k-i)]. \quad (11)$$

We can split $SM(n,k)$ into two parts, $SM(n,k) = [SM_+(n,k) + SM_-(n,k)]/2$, where:

$$SM_+(n,k) = HT^2(n,k) + 2\sum_{i=1}^{L_d} P(i)HT(n,k+i)HT(n,k-i), \quad (12)$$

$$SM_-(n,k) = HT^2(n,-k) + 2\sum_{i=1}^{L_d} P(i)HT(n,-k+i)HT(n,-k-i). \quad (13)$$

Knowing that $HT(n,-k) = HT(n,N-k)$, follows $SM_-(n,k) = SM_+(n,N-k)$, and consequently,

$$SM(n,k) = \frac{1}{2}[SM_+(n,k) + SM_+(n,N-k)]. \quad (14)$$

A. Signal Independent Window

In the case of signal independent window $P(i)$ width, the hardware for one channel implementation and $L_d = 2$ is presented in Fig. 3. It has been designed for a 16 bit fixed-point arithmetic. A system for $SM_+(n,k)$ realization is the similar to the one designed for $STFT_{\mathrm{Re}}(n,k)$ realization, and presented in Fig. 1b), with $STFT_{\mathrm{Re}}(n,k)$ being replaced by $HT(n,k)$. Keeping in mind (14), the second part of the SM, $SM_-(n,k)$ may be obtained from the $(N-k)$-th channel. It is important to note that the HT realization (BLOCK 1 in Fig. 3) is simpler than the STFT realization (BLOCK 1 in Fig. 1). The $SM(n,k)$ value is an average of $SM_+(n,k)$ and $SM_+(n,N-k)$, (14). Conclude that the SM implementation based on the HT needs only a half of the hardware used in the realization of BLOCK 2 in Fig. 1b), since the HT is always real, and $SM_-(n,k) = SM_+(n,N-k)$.

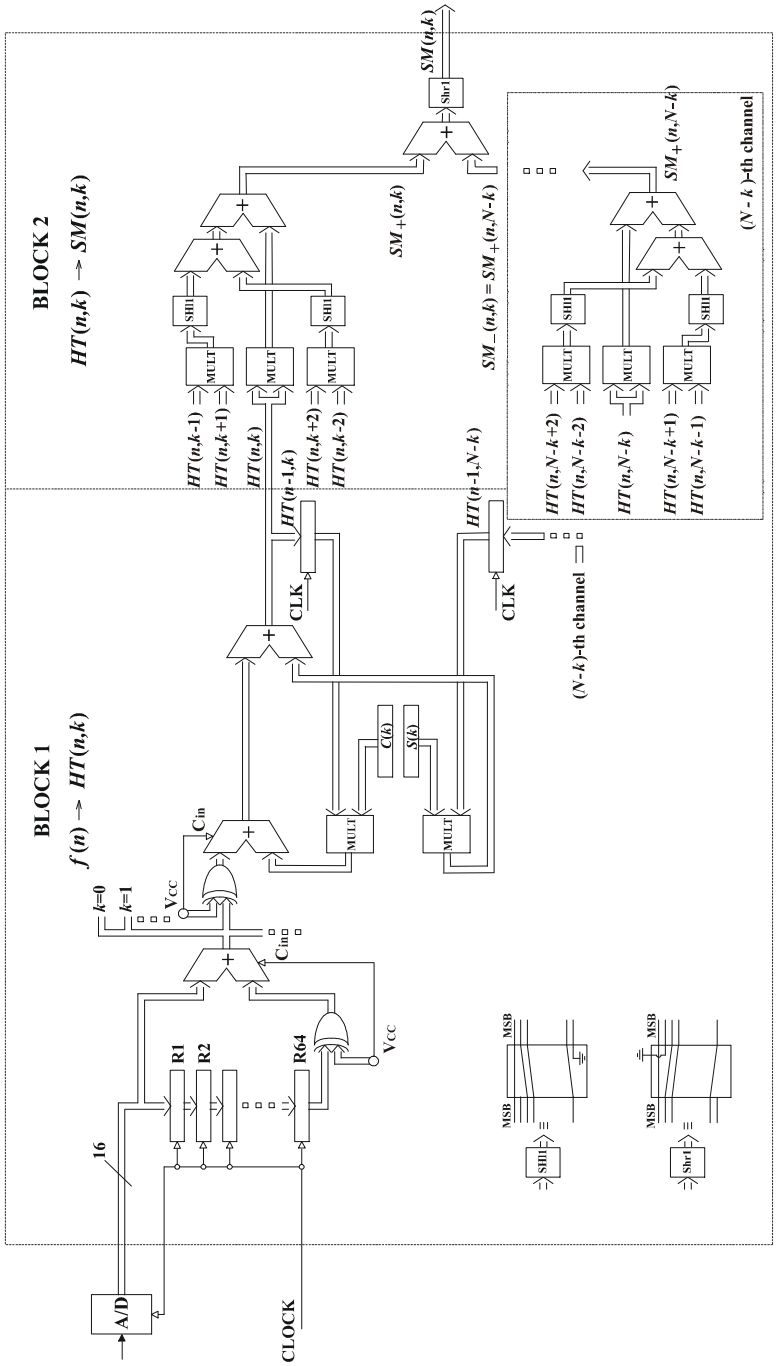

Fig. 3. Architecture for the signal independent S-method realization using the Hartley transform (one channel with $L_d = 2$).

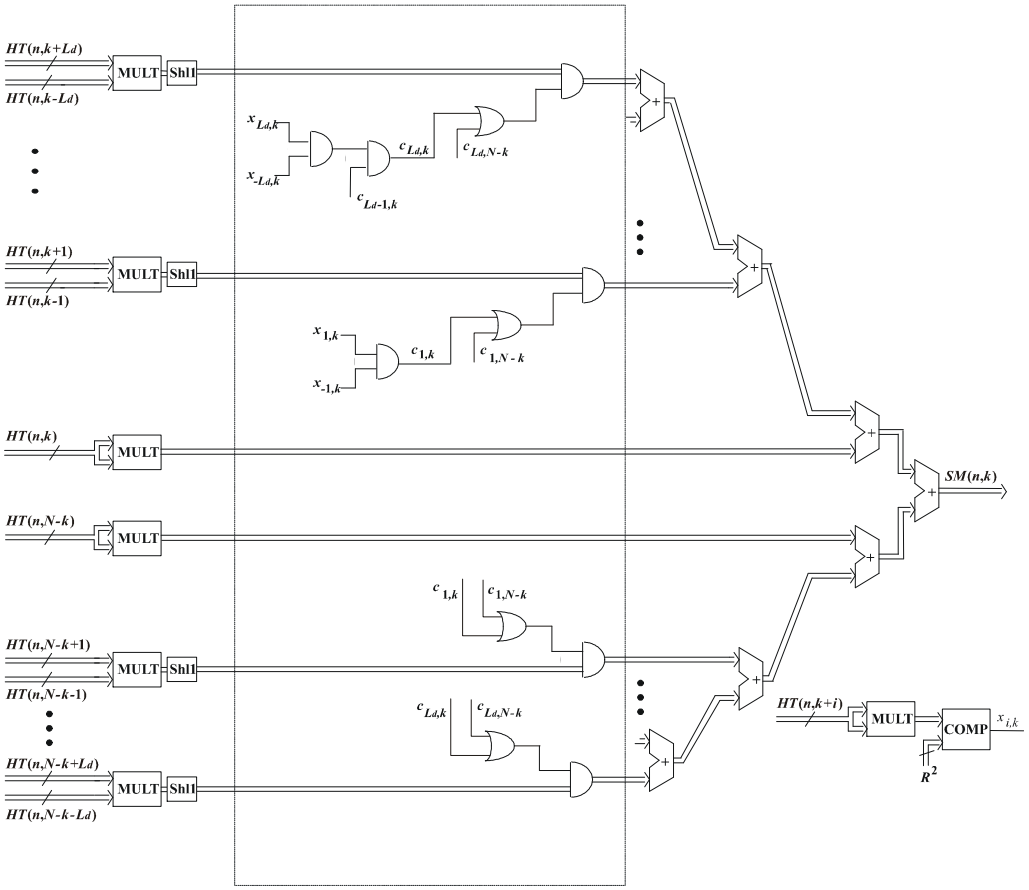

Fig. 4. Hardware modification for the signal dependent S-method realization using the Hartley transform.

B. Signal Dependent Window

For signal dependent realization we have to produce signals which will control the summation in $SM_+(n,k)$ and $SM_-(n,k) = SM_+(n,N-k)$. These signals should stop the summation (i.e., switch off lines in the hardware implementations) outside the auto-therm width, defined by the spectrogram. In order to obtain these values, note that the spectrogram is calculated based on HT, as:

$$SPEC(n,k) = \frac{1}{2}[HT^2(n,k) + HT^2(n,N-k)]. \tag{15}$$

It is equal to zero when both $HT(n,k)$ and $HT(n,N-k)$ are zeros at the same time. It means that the end of an auto-term is reached, and the summation should stop, if both $HT(n,k)$ and $HT(n,N-k)$ are zeros (or practically, below a reference value R). Control signals for expressions (12) or (13) separately, may be obtained from the system presented in Fig. 2, where the inputs are the HTs, $HT(n,k)$ or $HT(n,N-k)$. But, having in mind the previous analysis, control signals for both transforms, $HT(n,k)$ and $HT(n,N-k)$ should be the same and equal to a logic sum of the control signals obtained from $HT(n,k)$ and $HT(n,N-k)$, Fig. 4.

Note that in this case the reference value is defined with respect to the spectrogram value, as:

$$R = \frac{1}{Q} \times \max_k \{\sqrt{[HT^2(n,k) + HT^2(n,N-k)]/2}\}. \tag{16}$$

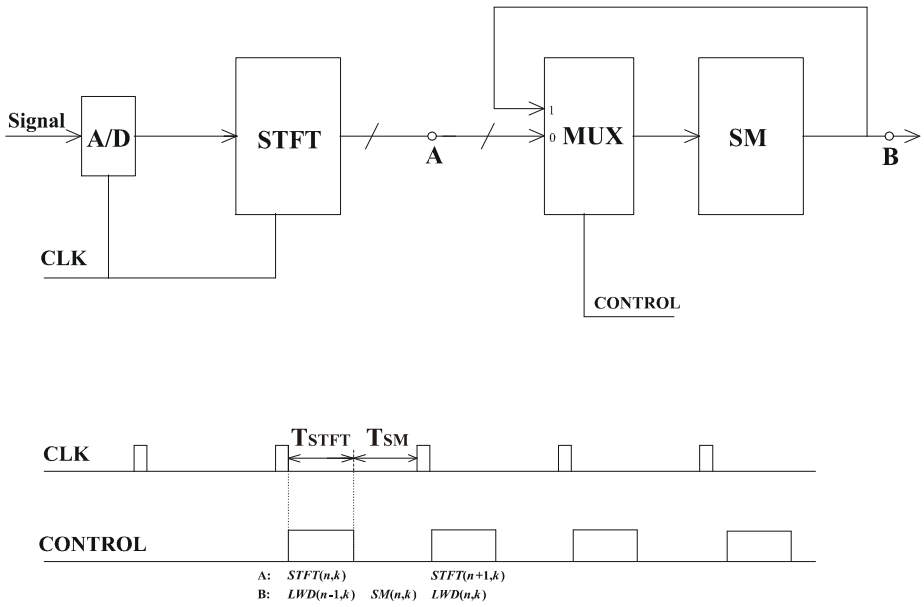

Fig. 5. Block scheme for the L-Wigner distribution realization, along with control signals.

IV. Hardware Implementation of the Higher Order Distributions

A. L-Wigner Distribution (L-WD)

Some characteristics of the TFDs can be improved by using higher order TFDs. For practical applications, especially interesting are those higher order TFDs which can be presented in the two-dimensional time-frequency plane, [7], [26], [30]. One of them, which may be very easily and efficiently realized using the proposed hardware is the L-WD, [30]. In the case of the nonlinear frequency modulated signals it can produce significantly higher concentration than the WD. Realization of the L-WD can be done in an order recursive manner:

$$LWD_L(n,k) = \sum_{i=-L_d}^{L_d} LWD_{L/2}(n,k+i)LWD_{L/2}(n,k-i), \tag{17}$$

where $LWD_L(n,k)$ is the L-th order L-WD and $LWD_1(n,k) = SM(n,k)$. This is the same form as (1). Therefore, it may be realized using the same hardware as in the SM realization. In order to share the existing hardware (presented in Fig. 1 and Fig. 3) for the realization of a system with $L = 2$, we have to introduce the multiplexers at the input of BLOCK 2. The multiplexer's control signal determines whether the system processes $STFT(n,k)$ or $SM(n,k)$. In the latter case only half of BLOCK 2 from Fig. 1 is used, since the SM is always real. During the first part of the sampling period (T_{STFT}) BLOCK 1 realizes $STFT(n,k)$, while BLOCK 2 calculates $LWD_2(n-1,k)$. In the second part of the sampling period (T_{SM}) control signal resets to zero, and $SM(n,k)$ is produced. Thus, for the L-WD realization we did not use any extra time, i.e. *with the same hardware, within the same time period, system produces two outputs: the SM and the L-WD with* $L = 2$. Since for $LWD_2(n-1,k)$ calculation only a half of BLOCK 2 (from Fig. 1) hardware is used, the other half may be used in order to simultaneously compute $LWD(n-2,k)$ for $L = 4$. The complete block diagram of the modified hardware, together with timing signals for multiplexer's control is presented in Fig. 5. Examples, illustrating the concentration improvement when the L-WD is used, are presented in [30].

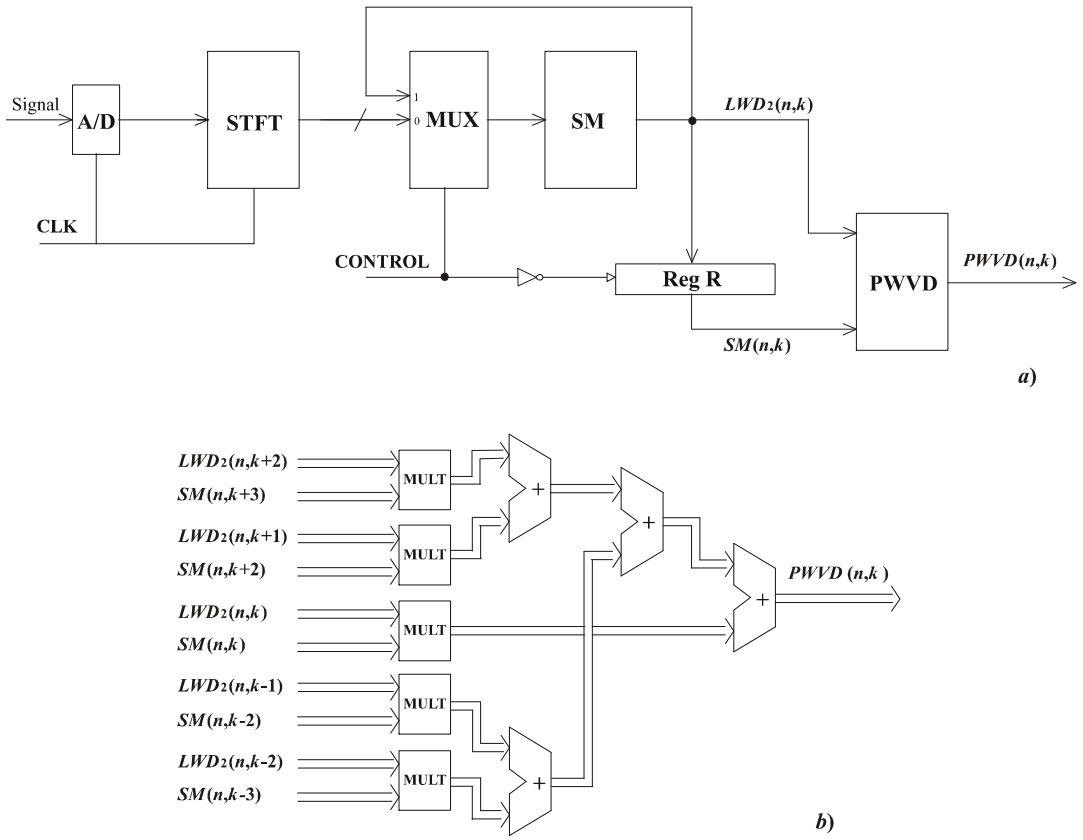

Fig. 6. Realization of the Polynomial Wigner-Ville distribution, *a*) Block scheme, *b*) Architecture for the PWVD block realization.

B. Polynomial Wigner-Ville Distribution (PWVD)

It is interesting to note that once we have obtained $SM(n,k)$ and $LWD(n,k)$, for $L=2$, we may easily get cross-terms free (reduced) fourth order PWVD, [7], [26], using the following relation, [33]:

$$PWVD(n,k) = \sum_{i=-L_d}^{L_d} LWD_2(n,k+i)SM(n,k+[i/A]),$$

where $A = 0.85/1.35$ is a constant and $[i/A]$ denotes the integer closest to i/A. Architecture for the realization of PWVD is presented in Fig. 6. Control signal is the same as in Fig. 5. In order to get a close approximation of $SM(n,k+i/A)$ (which is not available since i/A is not an integer) using $SM(n,k+[i/A])$, usually some interpolation of the SM will be needed, [7], [33]. More details are presented in [33].

V. Application to the Time-Varying Filtering

Analysis and processing of highly nonstationary signals can be improved by using joint time-frequency domain tools. They are based on the TFDs. Time-varying filtering is one of the challenging areas where one can benefit from the TFDs. Time-varying filtering of a nonstationary continuous-time noisy signal $x(t) = f(t) + \varepsilon(t)$, based on the WD, is defined as [9], [22], [28], [31]:

$$(Hx)(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} L_H(t,\omega) STFT_x(t,\omega) d\omega, \tag{18}$$

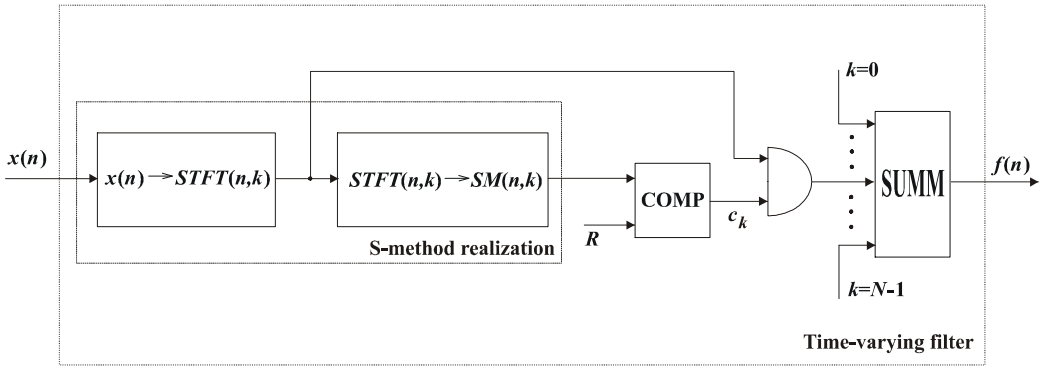

Fig. 7. Block scheme for hardware realization of time-varying filter.

where

$$L_H(t,\omega) = E\{WD_{fx}(t,\omega)\}/E\{WD_{xx}(t,\omega)\}$$

is the filters' region of support, [22], [28], [31], [34]. Note that subscripts are introduced in order to denote the TFDs of signal $x(t)$ at the filter's input, or the TFDs of the filtered signal $f(t)$, or their cross-TFDs.

Starting from the properties of the SM, and its simple hardware realization, it can be easily concluded that the usage of $SM(t,\omega)$ in $L_H(t,\omega)$ definition, in the case of multicomponent signal $f(t)$, is an appropriate approximation of $E\{WD(t,\omega)\}$, when only one noisy signal realization is known. Thus,

$$L_H(t,\omega) = \frac{SM_{fx}(t,\omega)}{SM_{xx}(t,\omega)}. \tag{19}$$

Besides, the SM implementation includes, as a key intermediate step, the STFT realization, which is included in the time-varying filtering definition (18), as well. Thus, the STFT realization may be used in both: the filter's region of support determination (by realization of the SM) and the filter implementation.

Consider a wide class of frequency modulated (highly concentrated in time-frequency plane) signals $f(t)$ corrupted with a white noise $\varepsilon(t)$, widely spread in the time-frequency plane. Knowing that the WD of this signal lies inside a tiny region D in the time-frequency plane, while the small part of noise lies inside this area (this part can be neglected with respect to the part of noise outside D), a simple solution for $L_H(t,\omega)$ may be given, [18]

$$L_H(t,\omega) = \begin{cases} 1, & (t,\omega) \in D \\ 0, & (t,\omega) \notin D. \end{cases} \tag{20}$$

Discrete form of (18) is given by, [31]:

$$(Hx)(n) = \sum_{k=-N/2+1}^{N/2} L_H(n,k) STFT_x(n,k). \tag{21}$$

In the hardware realization of the k-th channel, $L_H(n,k)$ can be determined by comparing the SM value with the spectral floor R, [34]. In that way we produce the control signals c_k, as:

$$c_k = \begin{cases} 1, & \text{for } SM_{xx}(n,k) \geq R \\ 0, & \text{for } SM_{xx}(n,k) < R, \end{cases} \tag{22}$$

that determine which component of $STFT(n,k)$ will be forwarded to the output. With a control signal $c_k = 1$, all components of the $STFT(n,k)$, $k = 0,1,...N-1$, are summed mutually. Block scheme of the time-varying filter hardware realization is presented in Fig. 7. The additions may be performed by adding the adjacent components in the first step, then the adjacent sums in the next step, and so on. That scheme corresponds to the butterflies in the FFT algorithms. Consequently, each component $STFT(n,k)$, $k = 0,1,...N-1$, with a control signal $c_k = 1$, passes through $\log_2 N$ adders to the output of the system.

In the hardware realization, Fig. 8, only the real part of STFT, $STFT_{\mathrm{Re}}(n,k) =$

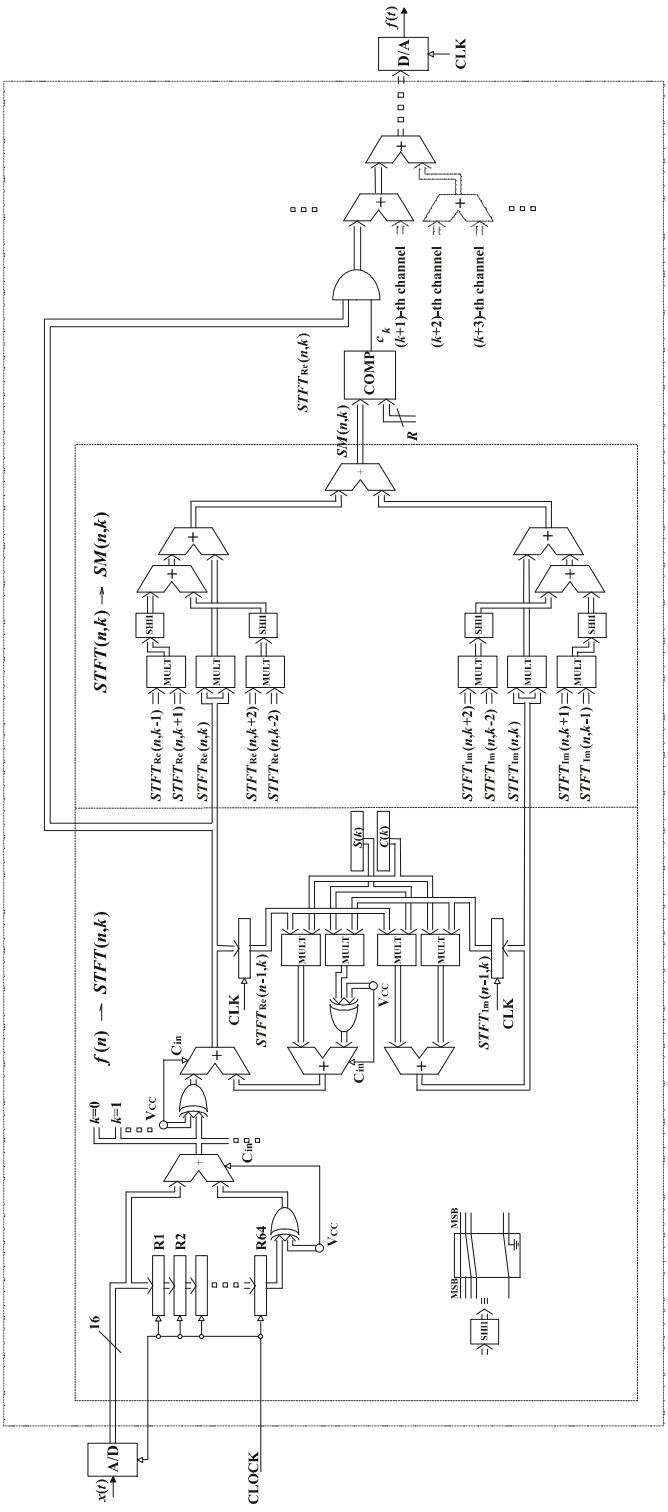

Fig. 8. A complete hardware for one channel realization of time-varying filter.

$\mathrm{Re}\{STFT_x(n,k)\}$, is used. Namely, for a symmetric function (in frequency) of the support region, $L_H(n,-k) = L_H(n,k)$, we have:

$$(Hx)(n) = \sum_{k=-N/2+1}^{N/2} L_H(n,k) STFT_{\mathrm{Re}}(n,k), \tag{23}$$

since the imaginary parts $STFT_{\mathrm{Im}}(n,k)$ vanish. Note that the symmetric function of $L_H(n,k)$ is obtained when the TFD (which is included in definition (19)) of real analyzed signal is the Hermitian function. It can be easily proved that the SM of a real signal satisfies this property, $SM(n,-k) = SM^*(n,k)$. Here, we will not present numerical illustration. For the examples we refer the readers to the papers where the theoretical approach for the methods used in this paper is given, [25], [30], [31], [33], [35]. Examples of time-varying filter application in hands-free telephone systems are presented in [34].

VI. Error Analysis

The use of finite word length registers in the implementation introduces two types of errors: signal quantization error and multiplication error. It is assumed that the signal is properly scaled to avoid an overflow. Quantization error can be modeled by an input additive noise. In this case we have $f(n) + \varepsilon(n)$ at the input instead of $f(n)$. Mathematically, this reduces to the well-known situation of noisy signals, [32] with TFD's variance:

$$\sigma^2 = \frac{N^2\sigma_\varepsilon^4}{2L_d+1}$$

$$+\frac{N\sigma_\varepsilon^2}{(2L_d+1)^2}\sum_{i=-L_d}^{L_d} SPEC_f(n,k+i), \tag{24}$$

where $\sigma_\varepsilon^2 = 2^{-2b}/12$ is the quantization noise variance, [23]. Note that for spectrograms of order 1 and $N\sigma_\varepsilon^2 \ll 1$ this expression is of order $N\sigma_\varepsilon^2/(2L_d+1)^2 = N2^{-2b}/(12(2L_d+1)^2)$. For example for $N = 64$, $b = 16$, and $L_d = 2$, σ^2 is of order 2^{-34}. The multiplication error $q(n)$ due to quantization of product $STFT(n-1,k)e^{j2\pi k/N}$, in (3), has the variance $\sigma_q^2 = 2^{-2b}/4$, [23]. This error accumulates with each iteration. Note that the proposed two-block solution has built in flexibility, which allows us to use any available hardware for the STFT realization. Namely, the BLOCK 1 from Fig. 1 can be implemented with the FFT based algorithms, instead of the recursive ones. In that case the accumulation error would be avoided and kept within the order of (24), [16]. More details on influence of finite register length effects to the obtained results are presented in [16].

VII. Hardware Realization and Implementation Results

There are many ways to implement the SM architecture. One of them is by using the DSP processors, running a special program. However, this solution is not so useful for real time processing, particularly at very high speeds. Another way is by using a specially designed ASIC or PD (Programmable Device). A great degree of parallelism at high speeds can be achieved with these architectures. To allow an economical and easy prototype development we chose FPGA (PD) instead of ASIC solution. Except the low cost, the main advantage of these devices is their "In-circuit reconfigurability" (ICR) that permits configuration on the board for the specific functionality. Additionally, the designer does not need to manage inventories of different ASIC designs.

Design of the circuit has been carried out through the serial and parallel forms realized in different FPGA chips from Altera, [13]. Both of them are implemented based on the architectures given in Fig. 1 (BLOCK 1). The AHDL (Altera Hardware Design Language) is used to model circuit components such as multiplexers, adders, multipliers and shift registers.

The serial architecture is realized according to the algorithm presented in Fig. 9. Its schematic diagram, including the components used, is given in Fig. 10. As can be seen, it has latency of 3 clock cycles per calculation (6 for real and imaginary part totally). The 16-bit STFT samples are fed to the inputs (pins) $D[0][15..0]$ to $D[4][15..0]$ simultaneously. During one clock cycle, two of them, via multiplexers MUX_1 and MUX_2, ad-

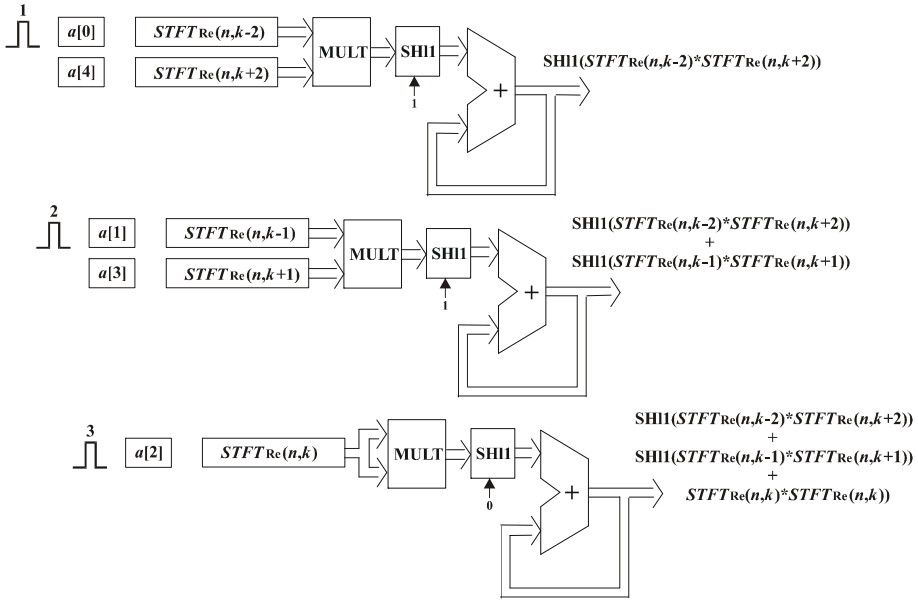

Fig. 9. Computation algorithm for serial architecture.

dressed by $ADD1[2..0]$ and $ADD2[2..0]$, are forwarded to the multiplier $MULT$, producing a 32-bit product on its outputs. After shifting, produced by the shift register $SHIFT\_REG$ ($DIS[4..0] = 1$ - left shift for 1, $DIS[4..0] = 0$ - no change) this product is passed to the adder's ($ADDER$) input "$dataa[\ ]$". Since the $datab[\ ] = Z[32..0]$, its output will be $Z[n][32..0] = Z[n-1][32..0] + data[n][32..0]$.

The serial architecture is realized in the device EPF10K30RC208-3, fabricated in CMOS SRAM technology, having 208 pins in RQFP package (141 available), 30.000 gates and 1.728 logic elements. It supports the speeds up to $100MHz$, consuming less than $0.5mA$ on $5V$. In the final implementation it requires 122 pins (86% of the available number), 87 inputs and 35 outputs, 1638 (94%) logic elements as well as 34 flip-flops. Additionally, total fan-in is 86% and synthesized cells ratio 36%.

In contrast to serial architecture, the parallel one has no latency. Its configuration is identical to the BLOCK 1's scheme, as shown in Fig. 11. It excludes the multiplexers, but increases the numbers of multipliers (totally 3), adders (2) and shift registers (3). For its practical realization a larger chip is required.

The parallel architecture is realized in larger device EPF10K100G503 from the same family. It has 503 pins in PGA package (400 available), 100.000 gates and 4.992 logic elements, supporting the speeds up to $100MHz$ on $5V$. In the final implementation 113 pins (28% of available number), 80 inputs, 33 outputs, 4338 logic elements (86%) and 0 flip-flops are required. Total fan-in is 75% and synthesized cells ratio 36%. In this case the number of the logic elements requires is increased about three times.

VIII. CONCLUSION

The systems for signal independent and signal dependent time-frequency analysis are presented. These systems use the STFT and the HT as an intermediate step, and produce better time-frequency signal representation than the spectrogram, regarding the most essential aspects, such as noise influence, cross-terms, oversampling and calculation complexity. Proposed hardware is very flexible and may simultaneously produce the SM, the L-WD and the PWVD. The same hardware can be used for time-varying filtering realization, as well. It is important to emphasize that additional time

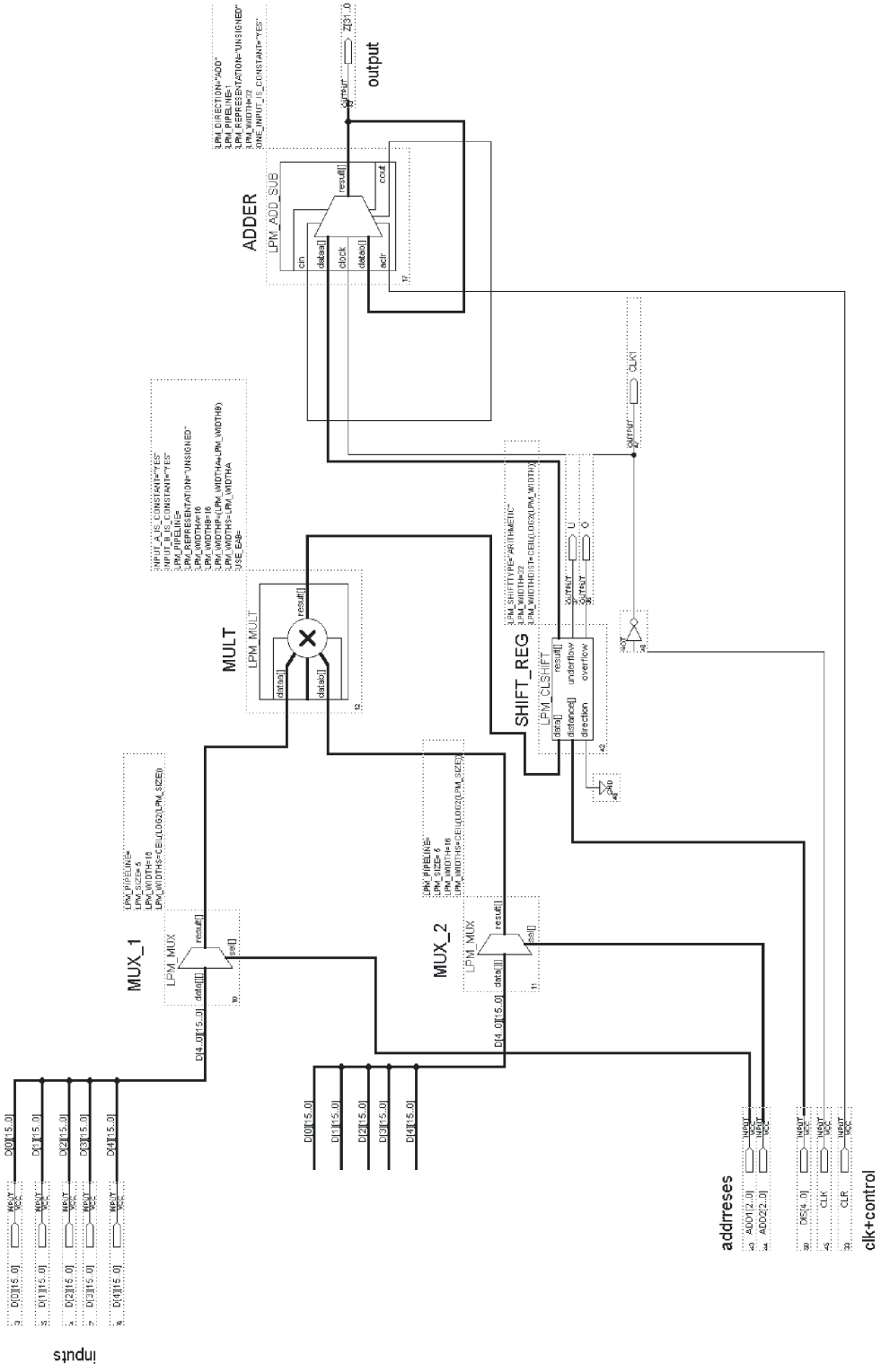

Fig. 10. A detailed scheme used for the chip realization in the serial configuration.

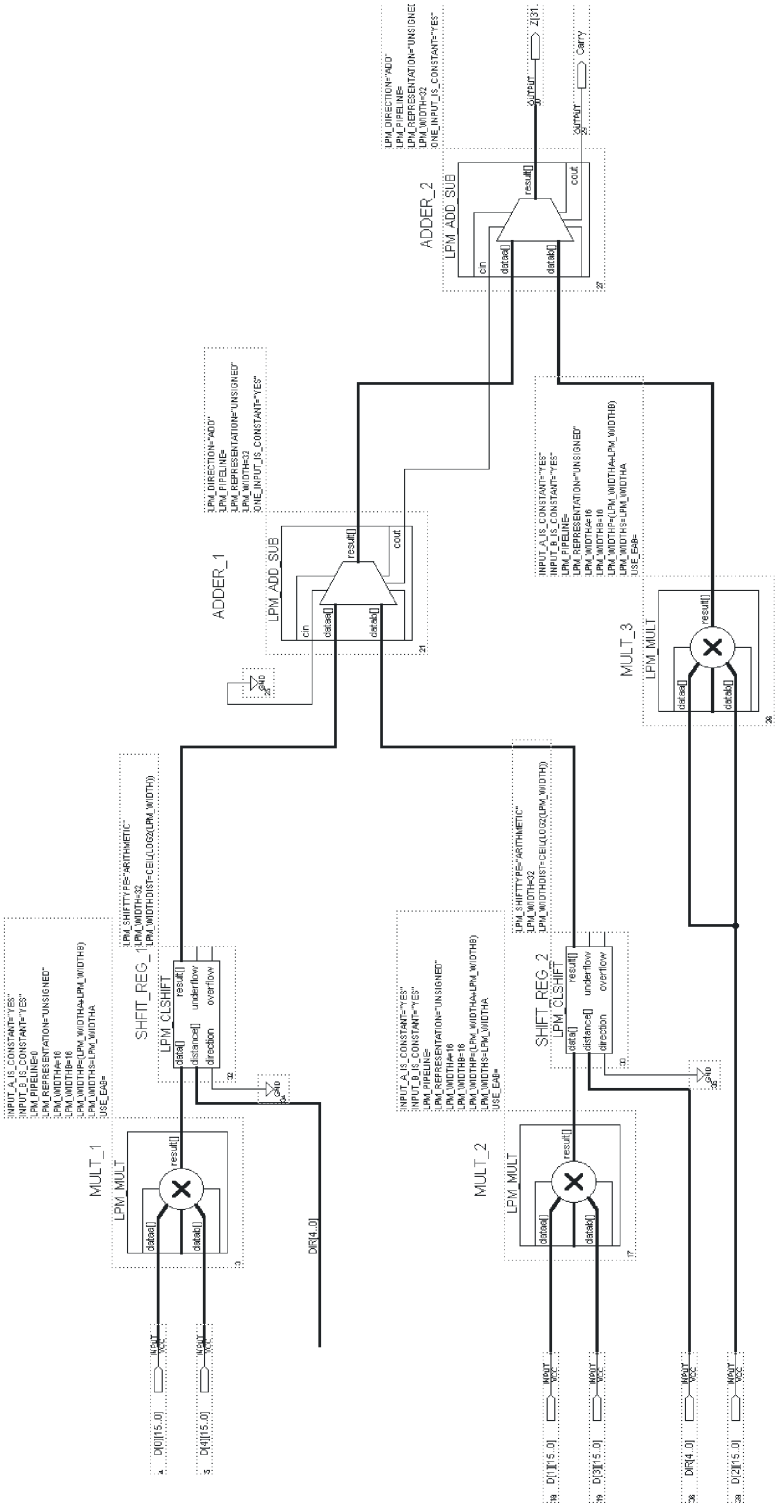

Fig. 11. A detailed scheme used for the chip realization in the parallel configuration.

requirements are not necessary when the designed hardware is used for the higher order TFDs implementation. Real-time realization of the FPGA chips for the SM implementation, in the serial and parallel configuration, are given.

IX. Acknowledgment

The author are thankful to the reviewer for very thorough comments that helped to improve the manuscript. This work is supported by the Volkswagen Stiftung, Federal Republic of Germany.

References

[1] *Proceedings of the IEEE* (1996), special issue on Time-Frequency Analysis, **84**, n° 9.

[2] Ackroyd (M.H.). Short time spectra and time-frequency energy distribution. *J. Acoust. Soc. Am.* (1970), **50**, pp. 1229-1231.

[3] Amin (M.G.). Spectral smoothing and recursion based on the nonstationarity of the autocorrelation function. *IEEE Trans. on Signal Processing* (1993), **41**, n° 2, pp. 930-934.

[4] Amin (M.G.). A new approach to recursive Fourier transform. *Proc. IEEE* (1987), **75**, pp. 1357-1358.

[5] Amin (M.G.), Feng (K.D.). Short time Fourier transform using cascade filter structures. *IEEE Trans. on Circuits and Systems-II* (1995), **42**, n° 10, pp. 631-641.

[6] Boashash (B.), Black (J.B.). An efficient real time implementation of the Wigner-Ville distribution. *IEEE Trans. Acoust., Speech, Signal Processing* (1987), **35**, n° 11, pp. 1611-1618.

[7] Boashash (B.), O'Shea (P.). Polynomial Wigner-Ville distributions and their relationship to time-varying higher order spectra. *IEEE Trans. on Signal Processing* (1994), **42**, n° 1, pp. 216-220.

[8] Boashash (B.), Ristić (B.). Polynomial time-frequency distributions and time-varying higher order spectra: Applications to analysis of multicomponent FM signals and to treatment of multiplicative noise. *Signal Processing* (1998), **67**, n° 1, pp. 1-23.

[9] Boudreaux-Bartels (G.F.), Parks (T.W.). Time-varying filtering and signal estimation using the Wigner distribution. *IEEE Trans. Acoust, Speech, Signal Processing* (1986), **34**, n° 6, pp. 442-451.

[10] Bracewell (R.N.). The fast Hartley transform. *Proc. IEEE* (1984), **72**, pp. 1010-1018.K.J.R.

[11] Cohen (L.). Time-frequency distributions - a review. *Proc. IEEE* (1989), **77**, pp. 941-981.

[12] Costa (A.H.), Boudreaux-Bartels (G.F.). A comparative study of alias-free time-frequency representations. *Proc. of the Int. Symp. TFTSA* (Philadelphia 1994), pp. 76-79.

[13] FLEX 10K Embedded Programmable Logic Device Family. *Altera Corporation, data sheet, Version 4.1* (2001).

[14] Goncalves (P.), Baraniuk (R.G.). Pseudo affine Wigner distributions: Definition and kernel formulation. *IEEE Trans. on Signal Processing* (1998), **46**, n° 6, pp. 1505-1517.

[15] Hlawatsch (F.), Broudreaux-Bartels (G.F.). Linear and quadratic time-frequency signal representation. *IEEE Signal Processing Magazine* (1992), pp. 21-67.

[16] Ivanović (V.), Stanković (LJ.), Petranović (D.). Finite register length effects in implementation of distributions from the Cohen class. *IEEE Trans. on Signal Processing* (1998), **46**, n° 7, pp. 2035-2041.

[17] Jonckheere (E.A.), Ma (C.). Split-radix fast Hartley transform in one and two dimension. *IEEE Trans. on Signal Processing* (1991), **39**, n° 2, pp. 499-503.

[18] Kozek (W.). Time-frequency signal processing based on the Wigner-Weyl framework. *Signal Processing* (1992), **29**, n° 1, pp. 77-92.

[19] Liu (K.J.R.). Novel parallel architectures for Short-time Fourier transform. *IEEE Trans. on Circuits and Systems-II* (1993), **40**, n° 12, pp. 786-789.

[20] Liu (K.J.R.), Chiu (C.T.). Unified parallel lattice structures for time-recursive discrete cosine/sine/Hartley transforms. *IEEE Trans. on Signal Processing* (1993), **41**, n° 3, pp. 1357-1377.

[21] Maharatna (K.), Dhar (A.S.), Banerjee (S.). A VLSI array architecture for realization of DFT, DHT, DCT and DST. *Signal Processing* (2001), **81**, n° 9, pp. 1813-1822.

[22] Matz (G.), Hlawatsch (F.), Kozek (W.). Generalized evolutionary spectral analysis and the Weyl spectrum of nonstationary random processes. *IEEE Trans. on Signal Processing* (1997), **45**, n° 6, pp. 1520-1534.

[23] Oppenheim (A.), Schafer (R.W.). Digital Signal Processing. *Prentice-Hall* (1975), pp. 404-464.

[24] Papoulis (A.). Signal Analysis. *McGraw-Hill Book Company* (1977).

[25] Petranović (D.), Stanković (S.), Stanković (LJ.). Special purpose hardware for time-frequency analysis. *Electronics Letters* (1997), **33**, n° 6, pp. 464-466.

[26] Ristic (B.), Boashash (B.). Relationship between the polynomial and higher order Wigner-Ville distributions. *IEEE Signal Processing Letters* (1995), **2**, n° 12, pp. 227-229.

[27] Scharf (L.L.), Friedlander (B.). Toeplitz and Hankel kernels for estimating time-varying spectra of discrete-time random processes. *IEEE Trans. on Signal Processing* (2001), **49**, n° 1, pp. 179-189.

[28] Shenoy (R.G.), Parks (T.W.). The Weyl correspondence and time-frequency analysis. *IEEE Trans. on Signal Processing* (1994), **42**, n° 2, pp. 318-331.

[29] Stanković (LJ.). A method for time-frequency analysis. *IEEE Trans. on Signal Processing* (1994), **42**, n° 1, pp. 225-229.

[30] Stanković (LJ.). A method for improved energy concentration in the time-frequency signal analysis using the L-Wigner distribution. *IEEE Trans. on Signal Processing* (1995), **43**, n° 5, pp. 1262-1268.

[31] Stanković (LJ.). On the time-frequency analysis

based filtering. *Annales des Telecommunications* (2000), **55**, n° 5-6, pp. 216-225.

[32] Stanković (LJ.), Ivanović (V.), Petrović (Z.). Unified approach to the noise analysis in the Wigner distribution and Spectrogram. *Annales des Telecommunications* (1996), **51**, n° 11-12, pp. 585-594.

[33] Stanković (LJ.), Stanković (S.), Djurović (I.). An architecture for realization of the cross-terms free polynomial Wigner-Ville distributions. *Proc. of the Int.Conf. ASSP* (Munich 1997).

[34] Stanković (S.). About time-variant filtering of speech signals with time-frequency distributions for hands-free telephone systems. *Signal Processing* (2000), **80**, n° 9, pp. 1777-1785.

[35] Stanković (S.), Stanković (LJ.). An architecture for the realization of a system for time-frequency analysis. *IEEE Trans. on Circuits and Systems-II* (1997), **44**, n° 7, pp. 600-604.

[36] Unser (M.). Recursion in short time signal analysis. *Signal Processing* (1983), **5**, n° 5, pp. 229-240.