

# Superior Execution Time Design of a Space/Spatial-Frequency Optimal Filter for Highly Nonstationary Two-Dimensional FM Signal Estimation

Veselin N. Ivanović, *Senior Member, IEEE*, Nevena R. Brnović

**Abstract**— Multiple-clock-cycle, signal adaptive, and fully pipelined hardware design of the optimal (Wiener) space/spatial-frequency filter is developed in this paper. All implementation and verification details, as well as the extensive comparative analysis, are provided. The developed solution optimizes critical design performances related to the hardware complexity, in line with multiple-clock-cycle nature. Variable (signal adaptive) number of clock cycles, taken within the execution in different space/spatial-frequency points, provides this solution to retain the optimized time requirements, as well as high resolution, selectivity, and estimation quality of the corresponding recently proposed signal adaptive filtering solution. However, as the major contribution, the fully pipelined implementation enables the developed design to additionally improve the time required for execution. The achieved improvement corresponds to a clock cycle per each space/spatial-frequency point performed within the estimation that results in the significant comparative improvement in execution time of up to 50% in terms of space/spatial-frequency points lying outside the local frequency of the estimated two-dimensional frequency-modulated signal. The implementation is tested on a highly nonstationary multicomponent signal and is verified by a field programmable gate array circuit design.

**Index Terms**— Execution time, Hardware design, Optimal filter, Pipelining, Space/Spatial-frequency representation.

## I. INTRODUCTION

EFFICIENT processing of nonstationary two-dimensional (2D) signals, including their filtering, requires space-varying approaches that can be defined by using space/spatial-frequency (S/SF) analysis tools (S/SF distributions), [1]-[5]. The S/SF analysis-based solutions are widely used, [6]-[13], and provide significant results in many practical applications,

such as the image filtering, [4], [6]-[9], texture segmentation, [10], [11], high resolution image analysis, [12], and optics, [13]. However, solutions based on the S/SF analysis require quite complex calculations and, therefore, significant time for execution, that seriously restrict their application in real-time, [1], [4]. Hardware implementations, especially with the optimized time requirements, in some cases, can overcome these problems, [14]-[18], as demonstrated in this paper.

A number of S/SF filtering solutions have been proposed so far, [1]-[5], [18]-[27]. Some of them, referred to as the classical ones (the Zadeh, Weyl, STFT and Gabor filters) or as their extended versions (the multiwindow STFT and Gabor filters, the minimum energy Weyl filter, as well as several forms of the halfband Weyl filter and approximate halfband Weyl filter), belong to the linear filtering solutions, [1]-[3]. Solutions whose regions of support are estimated based on the non-linear distributions belong to the non-linear ones, [4], [5], [18], [19], [21]-[23], [25], [26], that improve resolution and selectivity, but at the expense of complexity. Signal adaptive, multiple-clock-cycle hardware implementation (MCI) of an optimal (Wiener) space/spatial-frequency (S/SF) filter, based on the correspondence of its region of support to the local frequency (LF) of the estimated two-dimensional (2D) signal and on the S/SF analysis-based LF estimation, has conceptually been considered in [26] and has recently been implemented in [18]. This solution takes multiple, but variable–signal adaptive–number of clock cycles (CLKs) in different S/SF points within the execution. In this way, as an essential MCI solution, it optimizes critical design performances related to the hardware complexity, making it a suitable system for real-time and on-a-chip implementation. Also, as a signal adaptive solution, it optimizes the execution time, even in comparison to the corresponding single-clock-cycle implementation (SCI) approaches, [18]. Further, the signal adaptive solution [18] provides the highest quality LF estimation, high S/SF resolution, and therefore a very efficient estimation of nonstationary 2D frequency-modulated (FM) signals that occupy a significant range of frequencies and is corrupted by the high additive white noise. Considering critical design performances, this solution outperforms the other possible LF estimation-based S/SF filters designed in RTL (Register Transfer Level) design methodology, as well as the state-of-the-art online filtering solutions, [18], qualifying itself as an optimal solution in many practical applications.

Manuscript submitted on Nov. 11, 2017, revised on Jan. 30, Feb. 26, 2018.

Veselin N. Ivanović, Nevena R. Brnović are with the Electrical Engineering Department, University of Montenegro, 81000 Podgorica, MONTENEGRO (e-mails: [very@ac.me](mailto:very@ac.me), [nevenar@ac.me](mailto:nevenar@ac.me)).

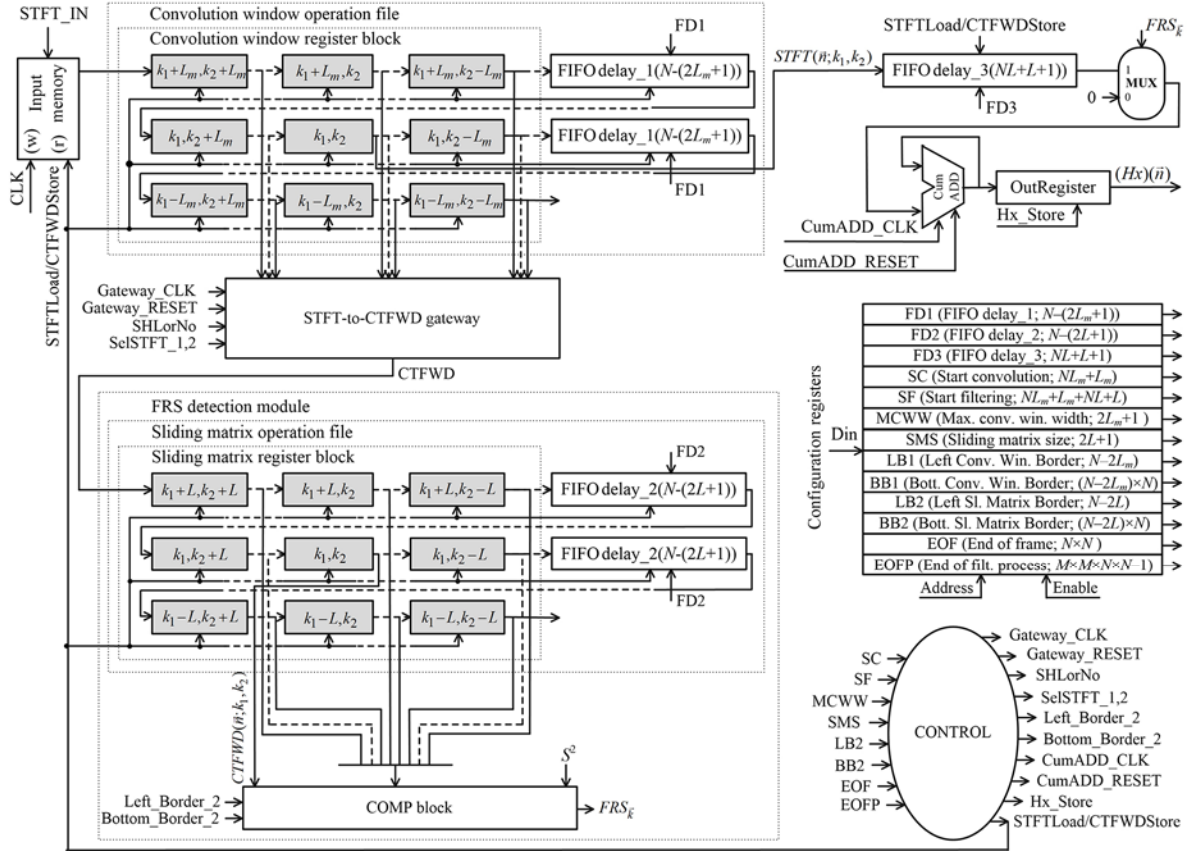


Fig. 1. Fully pipelined implementation of the optimal S/SF filter. Configuration registers contain stored parameters, their descriptions, and values, expressed by the number of needed  $STFT\_Load/CTFWD\_Store$  cycles, where  $M \times M$  is the 2D signal size.

However, as noted, the significant execution time, required by the systems based on the S/SF analysis, seriously restricts the application of these systems in real-time. Therefore, the development of solutions with the improved time requirements is of high importance. To this end, in this paper, the fully pipelined implementation of the LF estimation-based S/SF filter is developed, verified, and compared with the state-of-the-art filtering solutions, including the solution from [18]. The implementation developed here retains all noted desirable characteristics of the solution from [18], but substantially improves the time required for execution. It provides overlapping in execution, corresponding to one CLK per each S/SF point performed within the estimation, resulting in a significant comparative improvement in execution time of up to 50%, where the improvement of 50% is achieved in terms of S/SF points lying outside the LF of the estimated signal.

In this paper, after the Introduction, a brief background theory is given. The fully pipelined design of the LF estimation-based S/SF filter is developed in Section III and is verified through the implementation on a field programmable gate array (FPGA) device in Section IV. The comparisons are performed and the conclusions are derived in Section V.

## II. THEORETICAL BACKGROUND

S/SF filter, related to the 2D Wigner distribution framework and used to avoid distortion of an estimated nonstationary 2D FM signal, can be defined in the frequency-frequency domain and in vector notation as [4], [18], [20], [25], [26],

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

where  $STFT_x(\vec{n}, \vec{k}) = DFT_{\vec{m}}[w(\vec{m})x(\vec{n} + \vec{m})]$  is the 2D short-time Fourier transform (2D STFT) which contains information about the estimated q-component signal corrupted by the additive noise  $\varepsilon(\vec{n})$ ,  $x(\vec{n}) = \sum_{i=1, \dots, q} f_i(\vec{n}) + \varepsilon(\vec{n})$ ,

$L_H(\vec{n}, \vec{k})$  is the Weyl symbol used to denote the region of support (FRS) of the observed filter,  $DFT_{\vec{m}}[\cdot]$  is the operator of the discrete Fourier transform (DFT) in  $\vec{m}$ ,  $w(\vec{m})$  is a real-valued 2D lag window,  $N \times N$  is the windowed 2D signal  $w(\vec{m})x(\vec{n} + \vec{m})$  duration, and  $\vec{n} = (n_1, n_2)$ . Considering 2D FM signals  $f_i(\vec{n})$ ,  $i = 1, \dots, q$ , highly concentrated (in the S/SF area) around their LFs, a widely spread white noise  $\varepsilon(\vec{n})$ , and the optimal (Wiener) filter case, [3], [28], the FRS corresponds to the combination of LFs of signals  $f_i(\vec{n})$ , [4], [18], [20], [26]. Therefore, in the observed case and the S/SF framework, the FRS exists in frequency-frequency points  $\vec{k}_i, i = 1, \dots, q$  in which S/SF distribution of noisy 2D signal  $x(\vec{n})$  has local maxima and, in the case of a single realization of  $x(\vec{n})$ , can be estimated by, [4], [18], [25], [26], [29],

$$LF_i(\vec{n}) = \arg[\max_{\vec{k} \in Q_{\vec{k}_i}} CTFWD_x(\vec{n}, \vec{k})], \quad (2)$$

where  $Q_{\vec{k}_i}$  is the basic frequency-frequency region around

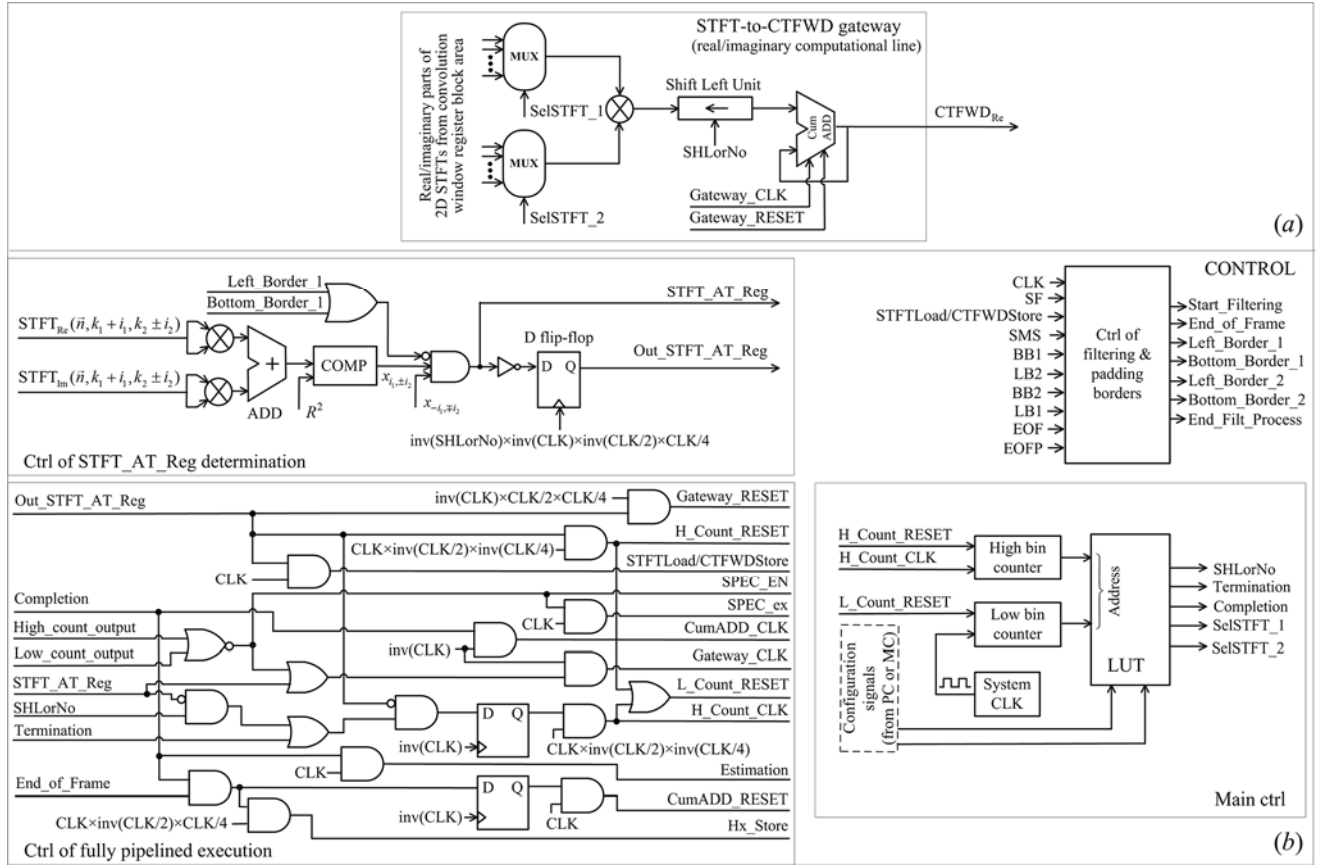


Fig. 2. (a) Real computational line of the STFT-to-CTFWD gateway, (b) CONTROL unit of the implementation from Fig. 1.

$f_i(\vec{n})$ , the LF of which is  $LF_i(\vec{n})$ , whereas  $CTFWD_x(\vec{n}, \vec{k})$  is the 2D cross-terms-free Wigner distribution (2D CTFWD), which has the best LF estimation characteristics among all S/SF distributions, [18], [25], [26], [29], [30], and, similar as (1), is the 2D STFT-based S/SF tool<sup>1</sup>, [17], [18], [31]. Besides,

<sup>1</sup> The 2D CTFWD, [17], [31], is defined to provide the 2D Wigner distribution-based S/SF representation of the analyzed 2D signal, but without the cross-terms presence. To this end, in an S/SF point  $(\vec{n}, k_1, k_2)$ ,  $k_1, k_2 = -N/2+1, \dots, N/2$ , it is defined as the limited 2D frequency-frequency convolution of the frequency-only-dependent 2D STFTs symmetrically distributed around the point in which the calculation is performed, [17], [31] (instead of the full frequency-frequency range 2D convolution used in the ordinary 2D Wigner distribution case, [4]). The convolution is limited by the rectangular 2D window of the signal adaptive width and centered at the S/SF point in which the 2D CTFWD is calculated. This window takes the minimal  $(1 \times 1)$  width in the S/SF points lying outside the 2D STFT auto-terms' domains, where the signal does not exist and where 2D STFTs of the non-noisy signal have zero value, while its maximum width  $(2L_m+1) \times (2L_m+1)$  can be taken only in the central S/SF point(s) of the widest 2D STFT auto-term(s)' domain(s). Therefore, the maximum convolution window width is determined by the widest 2D STFT auto-term(s)' domain(s) to provide the desired 2D Wigner distribution-based auto-terms representation. Simultaneously, it should not be greater than the minimal distance between different LFs of the analyzed 2D signal to provide cross-terms-free S/SF representation.

In practice, the 2D convolution window of maximum width  $(2L_m+1) \times (2L_m+1)$  slides over the frequency-only-dependent 2D STFTs. In each particular position, it creates the base for the 2D CTFWD calculation in S/SF point corresponding to the central convolution window element. However, in the calculation (the 2D frequency-frequency convolution of 2D STFTs grouped by the convolution window), only the 2D STFTs lying inside the 2D STFT auto-terms' domains are included (i.e., in practice and the noisy signal case, only the 2D STFTs whose squared absolute values are greater than the predefined reference level  $R^2$ , where  $R^2$  determines 2D STFT auto-

terms' domains). As the convolution window slides over the frequency-only-dependent 2D STFTs, the calculations in different S/SF points  $(\vec{n}, k_1, k_2)$ ,  $k_1, k_2 = -N/2+1, \dots, N/2$ , from the same signal point  $(n_1, n_2)$  are performed, as schematically presented in the left-hand side of Fig. 3. After completion of the calculation in each S/SF point from the observed signal point  $(n_1, n_2)$ , the same calculation procedure follows in the next signal point  $(n_1, n_2+1)$ , Fig. 5.

### III. FULLY PIPELINED DESIGN

The fully pipelined LF estimation-based optimal (Wiener) S/SF filter, described by (1) and (2) and developed here, is presented in Fig. 1, Fig. 2, and in Table 1. The implementation shown in Fig. 1 follows the same theoretical principles (see Section II) as the implementation designed in [18]. The novel control of the implementation, proposed and developed in this paper, Fig. 2(b), provides the unique, fully pipelined execution of the considered S/SF filter. The fully pipelined execution enables the proposed design to make crucial improvement in execution time and represents the major contribution given in this paper, described in detail in this Section and tested, proven, and verified in Section IV. To specify the control, the design technique based on the finite state machine is used. This technique provides clear graphical description and explanation of the execution either in a small finite state

terms' domains). As the convolution window slides over the frequency-only-dependent 2D STFTs, the calculations in different S/SF points  $(\vec{n}, k_1, k_2)$ ,  $k_1, k_2 = -N/2+1, \dots, N/2$ , from the same signal point  $(n_1, n_2)$  are performed, as schematically presented in the left-hand side of Fig. 3. After completion of the calculation in each S/SF point from the observed signal point  $(n_1, n_2)$ , the same calculation procedure follows in the next signal point  $(n_1, n_2+1)$ , Fig. 5.



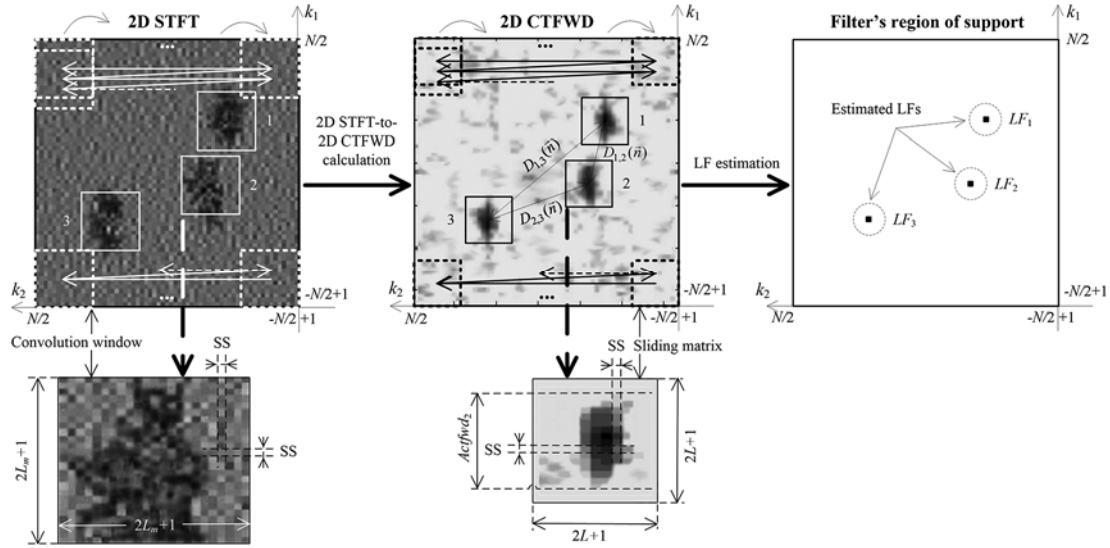


Fig. 3. Schematic presentation (in the frequency-frequency plane and for a particular signal point  $(n_1, n_2)$ ,  $n_1, n_2 = -M/2+1, \dots, M/2$ ) of the 2D STFT-to-2D CTFWD generation and of the LF estimation for the noisy signal (4) case. The sliding procedures of the convolution window and of the sliding matrix, that additionally determine order of the S/SF points performed within the execution in a particular signal point, are also presented. SS denotes sliding step of the convolution window and of the sliding matrix in each frequency direction, whereas  $D_{i,j}(\vec{n}) = |LF_i(\vec{n}) - LF_j(\vec{n})|$ .

machine case or in the case of a finite state machine including similar states with the same or very similar outputs, [32]. The latter is the case considered in this paper. In addition, this technique provides the control to be represented in a form that allows the detailed implementation to be synthesized on an integrated device, [32], such as FPGA one.

In Fig. 1, the implementation is presented for a predefined maximum 2D convolution window width  $(2L_m+1) \times (2L_m+1)$  and the sliding matrix size  $(2L+1) \times (2L+1)$ , where  $L_m$  and  $L$  values are determined by the 2D CTFWD definition, [17], [31], footnote 1, and by the LF estimation procedure<sup>2</sup>, [18],

<sup>2</sup> Procedure proposed in [18] and used as a development base in this paper provides high quality LF estimation based on the 2D CTFWD signal representation. To this end, the 2D matrix of size  $(2L+1) \times (2L+1)$  slides over the frequency-only-dependent 2D CTFWDs (see central part of Fig 3) in the same way as the 2D convolution window slides within the 2D CTFWD calculation, but over the frequency-only-dependent 2D STFTs (footnote 1). Opposed to the convolution window function, by grouping the corresponding 2D CTFWDs in each particular sliding position, the 2D matrix creates a basic frequency-frequency region from (2), while the LF existence is tested in S/SF point corresponding to the 2D matrix central element. In line with (2), an LF is detected only if the central one is the maximum sliding matrix element and if it is simultaneously greater than the predefined spectral level  $S^2$  (to determine the 2D CTFWD auto-terms' domains and to suppress noise influence outside them). To minimize both the estimation error inside 2D CTFWD auto-terms' domains and the negative influence of the frequency discretization, [4], and, therefore, to provide high quality LF estimation, the sliding matrix size  $(2L+1) \times (2L+1)$  should be determined by the widest 2D CTFWD auto-term(s)' domain(s), [18]. However, it should simultaneously be limited by the minimal distance between different LFs of the analyzed 2D signal to provide the LF estimation in the multicomponent signal case, [18]. In the other words,

$$\max_{1 \leq i \leq q} \{Actfwd_i\} \leq 2L+1 < \sqrt{2} \cdot \min_{1 \leq i, j \leq q, i \neq j} |LF_i(\vec{n}) - LF_j(\vec{n})|. \quad (3)$$

$Actfwd_i$ ,  $i=1, \dots, q$ , are widths of the non-overlapping 2D CTFWD auto-terms' domains, while  $|LF_i(\vec{n}) - LF_j(\vec{n})|$  denotes distance (in a frequency-frequency plane) between different LFs,  $LF_i(\vec{n})$  and  $LF_j(\vec{n})$ ,  $i, j=1, \dots, q$  and  $i \neq j$ , Fig. 3.

In hardware realizations, spectral level  $S^2$  is selected as a few percent of the 2D CTFWD's maximum value and based on the *a priori* knowledge about the estimated signal's range, determined by the optimal usage of the A/D

converter and memory locations, [17]. Spectral level  $R^2$ , used within the 2D CTFWD calculation (footnote 1), is selected in the same way, but relatively considered regarding the maximum value of the squared absolute 2D STFT.

Range (3) of possible sliding matrix sizes allows the estimation as long as 2D CTFWD auto-terms' domains do not occupy wider frequency-frequency ranges than the minimal distance between different LFs, Fig. 3. In this way and based on the fact that the 2D CTFWD provides maximum concentration of the analyzed 2D FM signal's components (footnote 3), the estimation in the non-overlapping highly nonstationary multicomponent signals case is enabled.

From the hardware development aspect, it is especially important that, per an S/SF point, the considered procedure requires only quite simple comparison of 2D CTFWDs, grouped by the sliding matrix, and of the spectral level  $S^2$ . Thus, as explicitly proven in [18], this procedure significantly improves calculation complexities of the state-of-the-art estimation algorithms, such as the algorithm from [4], as well as the 2D LMS and 2D RLS algorithms, [34].

converter and memory locations, [17]. Spectral level  $R^2$ , used within the 2D CTFWD calculation (footnote 1), is selected in the same way, but relatively considered regarding the maximum value of the squared absolute 2D STFT.

Range (3) of possible sliding matrix sizes allows the estimation as long as 2D CTFWD auto-terms' domains do not occupy wider frequency-frequency ranges than the minimal distance between different LFs, Fig. 3. In this way and based on the fact that the 2D CTFWD provides maximum concentration of the analyzed 2D FM signal's components (footnote 3), the estimation in the non-overlapping highly nonstationary multicomponent signals case is enabled.

From the hardware development aspect, it is especially important that, per an S/SF point, the considered procedure requires only quite simple comparison of 2D CTFWDs, grouped by the sliding matrix, and of the spectral level  $S^2$ . Thus, as explicitly proven in [18], this procedure significantly improves calculation complexities of the state-of-the-art estimation algorithms, such as the algorithm from [4], as well as the 2D LMS and 2D RLS algorithms, [34].

to implement the LF estimation (2) and the fully pipelined S/SF filtering execution (1). Sliding matrix register block area implements basic frequency-frequency region from (2) and provides inputs of the COMP block. Implementing (2), through a set of 2-input comparisons combined with the basic logic operations, the COMP block tests an LF existence in the S/SF point corresponding to the central sliding matrix register block element and lying outside the bordering positions,  $\text{inv}(\text{Left\_Border\_2} + \text{Bottom\_Border\_2}) = 1$ . FIFO delay\_2 blocks provide sliding of the sliding matrix register block area over the calculated 2D CTFWDs, Fig. 3, to enable the COMP block to test the LF existence in different S/SF points  $(\vec{n}, \vec{k})$ ,  $n_1, n_2 = -M/2 + 1, \dots, M/2$ ,  $k_1, k_2 = -N/2 + 1, \dots, N/2$ . Moreover, FIFO delay\_1 blocks in combination with the convolution window register block and FIFO delay\_2 blocks in combination with the sliding matrix register block create shift memory structures, managed by the *STFTLoad/CTFWDStore* cycle and named the Convolution window operation file and the Sliding matrix operation file, respectively. The time of storing data and the time of propagation data, between the adjacent locations inside each of these structures, correspond to the time of storing data into a single register. The LF existence, determined by the unity  $FRS_{\vec{k}}$  value, allows the FIFO\_delay\_3 output element, correspondent in frequencies to the central sliding matrix register block element, to participate, based on (1), in the output signal  $(Hx)(\vec{n})$  generation.

Input 2D STFT data are imported to the proposed design on every CLK, since, as will be seen, in the proposed design this period essentially corresponds to the minimal execution time per an S/SF point. However, within the execution in S/SF points lying inside the 2D STFT auto-terms' domains, a greater number of CLKs is required (see Fig. 9). Therefore, the input memory is introduced in Fig. 1 to enable the design to propagate the imported data through the system by the signal adaptive *STFTLoad/CTFWDStore* cycle. As discussed, to provide the LF estimation-based S/SF filtering, the proposed design respectively performs the 2D CTFWD calculation [31] and the LF estimation in each particular S/SF point. To this end, within the execution in different S/SF points, the 2D CTFWD calculation takes multiple, but variable (signal adaptive) number of CLKs, determined by the *STFTLoad/CTFWDStore* cycle, while in each of these points, the LF estimation is performed in the separate-*Estimation*-CLK. A number of taken CLKs depends on the relative position of the point in which the calculation is performed regarding the 2D STFT auto-terms' domains and corresponds to the total number of 2D STFTs pairs that participate in convolution within the 2D CTFWD calculation (footnote 1) increased for the *Estimation* CLK. Thus, CLK time of the implementation should provide the calculation corresponding to the minimal convolution window width of  $(1 \times 1)$ , when only one 2D STFTs pair is included in the calculation. Within the execution in a particular S/SF point, the first execution CLK (the *SPEC execution* one) in combination with the *Estimation* one provide the S/SF filtering based on the LF estimation and on the 2D STFT energetic form (2D Spectrogram). Therefore, these CLKs are the unconditional ones. Residual CLKs improve the filtering quality up to the 2D CTFWD-based one,

but these CLKs are conditional ones because they can be performed only in S/SF points lying inside the 2D STFT auto-terms' domains<sup>3</sup>, Fig. 8, Fig. 9. Note, partial results achieved in each CLK performed within the 2D CTFWD calculation in an S/SF point are saved in the gateway's CumADD.

To control the LF estimation-based S/SF filtering execution, the *STFT\_AT\_Reg* signal is generated in Fig. 2(b) as a basic signal adaptive one. It is recalculated (for different 2D STFTs pairs) in each particular CLK performing within the execution in an S/SF point, to define the relative position of the observed point regarding the 2D STFT auto-terms' domains. The *STFT\_AT\_Reg* signal (its zero value) allows the *SHLorNo* signal, set in the Look-up-Table (LUT) memory, Table 1, to terminate the 2D CTFWD calculation [31] in the CLK in which the calculation exits the 2D STFT auto-terms' domains. The 2D CTFWD calculation is completed in the next-*Estimation*-CLK. To this end, the *Out\_STFT\_AT\_Reg* signal, generated based on the inverted *STFT\_AT\_Reg* signal value, Fig. 2(b), participates in the *STFTLoad/CTFWDStore* signal creation, Fig. 2(b), to allow the calculated 2D CTFWD sample to be stored in the first register of the sliding matrix register block area, Fig. 1, that additionally results in shifting of this area (over the calculated 2D CTFWDs) for one position right, Fig. 3, and, then, in the LF estimation within the first half of the *Estimation* CLK. In parallel, the next 2D STFT sample is imported to the convolution window register block area and this area is shifted (over input 2D STFTs) for one position right. After that and after the STFT-to-CTFWD gateway reset (performed by the *Gateway\_RESET* signal), the execution for the next S/SF point can begin. The fully pipelined execution will enable the design to begin the execution in the next S/SF point (with the *SPEC execution* CLK) in parallel with the *Estimation* CLK of the previous S/SF point. In this way, the *STFT\_AT\_Reg* signal essentially controls the signal adaptive number of CLKs taken per an S/SF point within the execution. Signals *Gateway\_CLK*, *CumADD\_CLK*, and *H\_Count\_CLK*, created by the CONTROL unit, Fig. 2(b), control operations of the STFT-to-CTFWD gateway, output CumADD, and High binary counter, respectively, while the *Gateway\_RESET*, *CumADD\_RESET*, and *H\_Count\_RESET* signals reset these modules. Main CLK controls operation of the Low binary counter, while the *L\_Count\_RESET* signal resets it.

Within the second half of the *Estimation* CLK, the FIFO delay\_3 output sample participates in the output signal calculation, but only if the LF is detected in the first half of the same CLK. Partial results in the output signal calculation achieved within the execution in each particular S/SF point

<sup>3</sup> In line with the definition (see footnote 1), the 2D CTFWD is introduced to provide both the maximum concentrated (2D Wigner distribution-based) signal representation in S/SF points lying inside 2D STFT auto-terms' domains and the cross-terms-free 2D Spectrogram-based signal representation outside these domains, [31]. Therefore, determination of the relative position of the S/SF point in which the 2D CTFWD element is calculated with respect to the 2D STFT auto-terms' domains has the crucial significance within the 2D CTFWD calculation. From the other side, as discussed in footnote 2, within the LF estimation implemented here, the 2D CTFWD auto-terms' domains has the similar significance. Besides, the sliding matrix register block area size  $L$ , given by (3), is determined by the widest 2D CTFWD auto-term's domain, while the maximum convolution window register block area width  $L_m$  is determined by the widest 2D STFT auto-term's domain, [31], footnote 1. Hence, since the 2D CTFWD significantly improves 2D STFT concentration (see Fig. 3), the  $L$  and  $L_m$  values should be selected such that  $L < L_m$ .

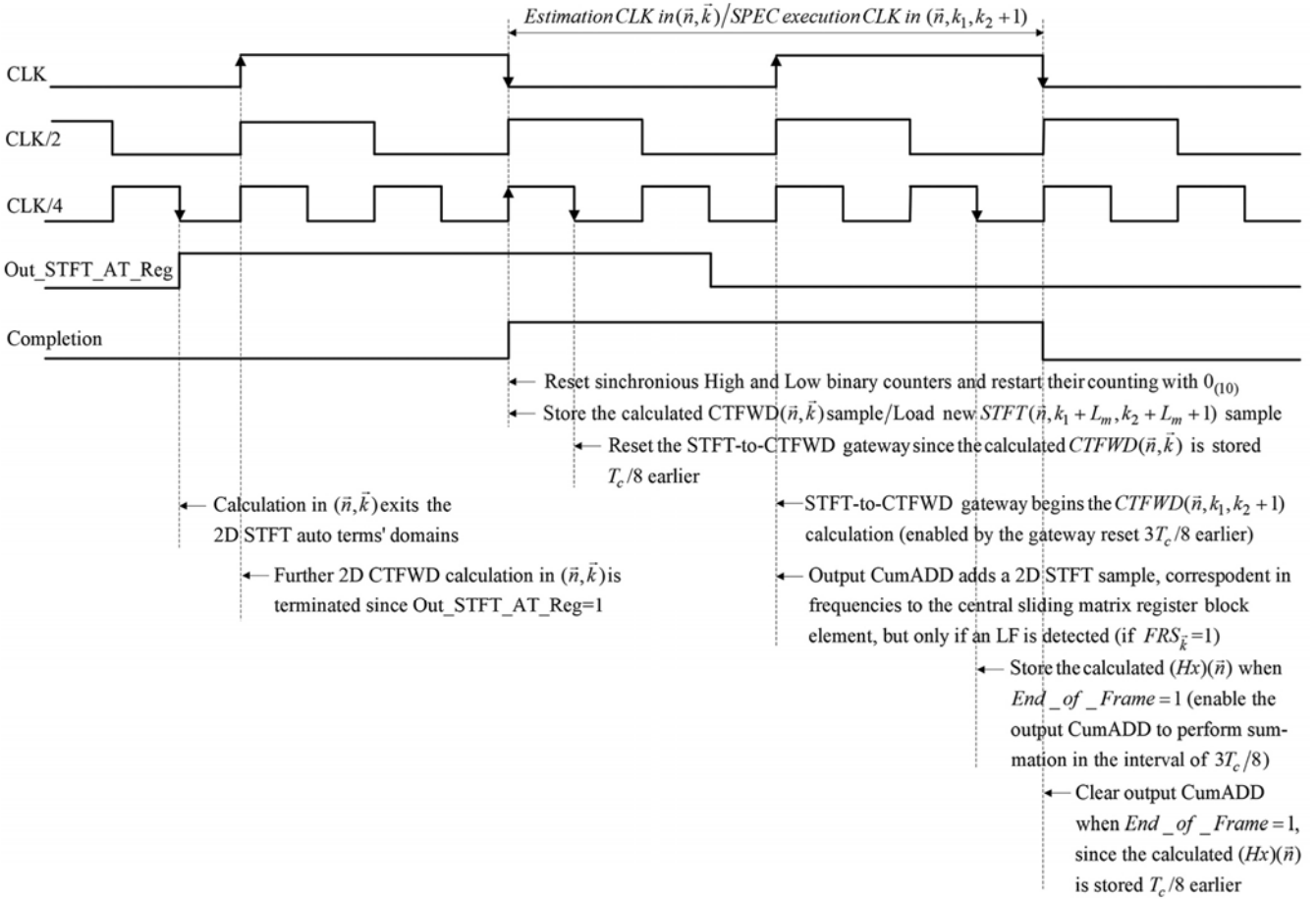


Fig. 4. The timing of operations that participate in execution within the *Estimation* and the *SPEC execution* CLKs respectively performed in the adjacent S/SF points, as well as within the potential *Completion* CLK. Instants at which the proposed design is triggered are additionally described, whereas  $T_c$  is the CLK time.

from the observed frame/signal point are saved in the output CumADD. After the execution in the maximum frequencies S/SF point, the final value of the output signal is obtained at the CumADD output, that is detected by the *End\_of\_Frame* signal creation to allow the *Completion* signal to complete the calculation (by the  $(Hx)(\bar{n})$  storage in the OutRegister) in the final, *Completion* CLK of a frame/signal point.

**Fully Pipelined Execution.** To provide participation in the other control signals creation, [18], [34], the *STFT\_AT\_Reg* signal is generated (through a multiplier, an adder, and a comparator, Fig. 2(b)) in a half of a CLK. Its generation defines both the longest path of the implementation and the fastest CLK time  $T_c$ ,  $T_c/2 = T_m + T_a + T_{comp}$  ( $T_m$ ,  $T_a$ ,  $T_{comp}$  – multiplication, addition, and comparison times, respectively), [18], [34]. However, operations that participate in execution within the *SPEC execution* and *Estimation* CLKs in each S/SF point (storing and propagation through the convolution window and sliding matrix operation files, gateway reset, and summation into the output CumADD) require significantly smaller time for their executions, as well as operations that create the unconditional, final-*Completion*-CLK of a frame/signal point ( $n_1, n_2$ ) (storing the calculated  $(Hx)(\bar{n})$  into the OutRegister and the output CumADD reset). In detail,  $T_c/8$  is more than sufficient time for storing and propagation through the shift memory structures, such as the convolution window and sliding matrix operation files, for the gateway and

output CumADD reset, and for storing into the OutRegister, whereas the interval of  $3T_c/8$ , assumed for the summation into the output CumADD, enables system realization until  $T_a \leq 3(T_m + T_{comp})$  is satisfied. Following these principles, to optimize the execution, the timing of above noted operations should be managed not only by the main CLK, but by its combinations with the twice faster (CLK/2) and fourfold faster (CLK/4) cycles, as implemented in Fig. 2(b), graphically supported and additionally explained in Fig. 4, and as will be verified by the real timing diagram presentation in Fig. 8. Cycles CLK, CLK/2, and CLK/4 are simply free-running signals with the fixed cycle times, generated in the corresponding mutually synchronized oscillators. The main CLK controls the execution in a particular S/SF point, as performed in [16], [18], whereas cycles CLK/2 and CLK/4 are introduced to provide additional control of the execution within the following CLKs: *SPEC execution*, *Estimation*, *Completion*. In line with the implementation given in Fig. 2(b) and the timing shown in Fig. 4, the fastest cycle CLK/4 in combinations with cycles CLK and CLK/2 provide timing of the gateway reset, Fig. 1, timing of the completion of the execution in a frame/signal point (storage of the calculated output signal in the OutRegister, Fig. 1), and timing of the *Out\_STFT\_AT\_Reg* signal creation, Fig. 2(b). In this way, the design developed here allows overlapping in execution between the adjacent S/SF points and in that process includes all unconditional CLKs performed within the estimation (the *SPEC execution*



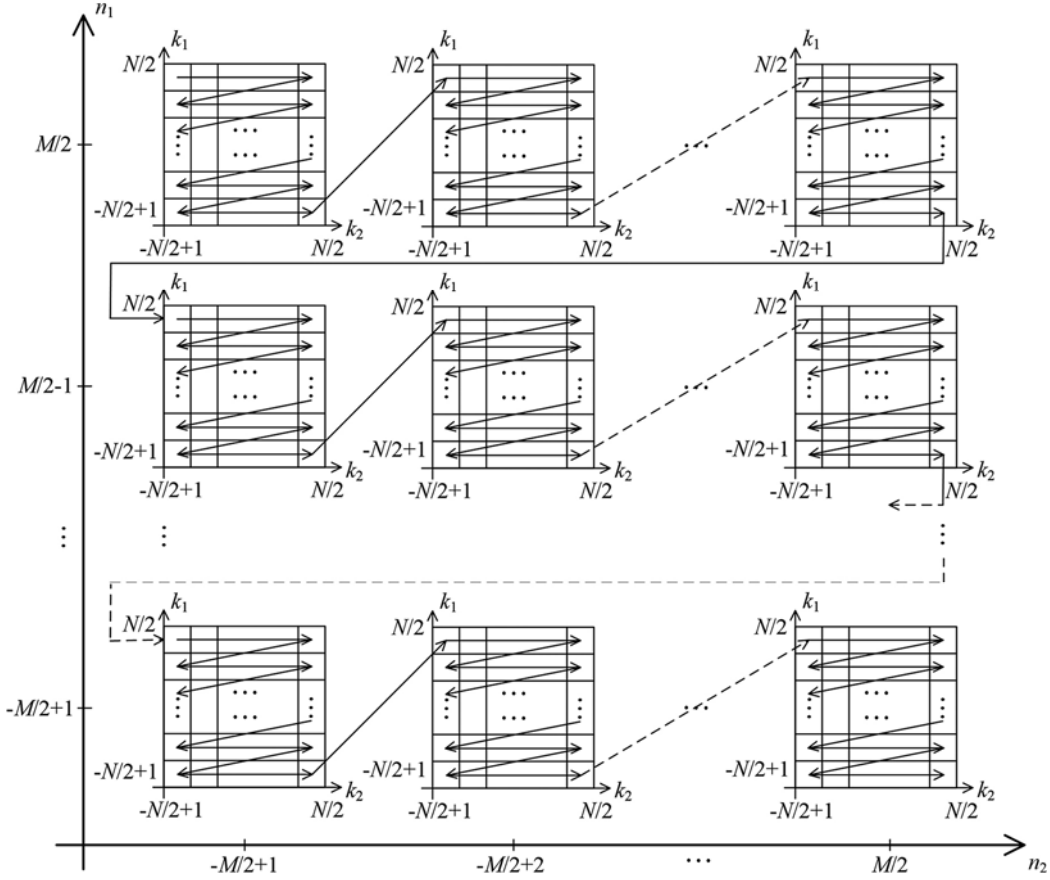


Fig. 5. S/SF points performed within the S/SF-based execution, their distribution in the 4-dimensional S/SF space, and the order of their execution.

and the *Estimation* CLKs of each S/SF point, as well as the *Completion* CLK of each frame/signal point), providing the fully pipelined execution and making essential improvement in comparison to [18]. In detail,

1. The *Estimation* and the *SPEC execution* CLKs, respectively performing in the adjacent S/SF points are mutually overlapped in execution, where within the S/SF-based execution, Fig. 5, the adjacent S/SF points are:

- S/SF points  $(\bar{n}, k_1, k_2)$  and  $(\bar{n}, k_1, k_2 + 1)$ ,  $n_1, n_2 = -M/2 + 1, \dots, M/2$ ,  $k_1, k_2 = -N/2 + 1, \dots, N/2$ , from the same frame/signal point  $(n_1, n_2)$  and the same frequency direction  $k_1$ , but the adjacent frequency directions  $k_2$  and  $k_2 + 1$ ,
- Bordering S/SF points  $(\bar{n}, k_1, N/2)$  and  $(\bar{n}, k_1 + 1, -N/2 + 1)$ ,  $n_1, n_2 = -M/2 + 1, \dots, M/2$ ,  $k_1 = -N/2 + 1, \dots, N/2$ , from the same frame/signal point  $(n_1, n_2)$ , but the adjacent frequency directions  $k_1$  and  $k_1 + 1$ ,
- Bordering S/SF points  $(n_1, n_2, N/2, N/2)$  and  $(n_1, n_2 + 1, -N/2 + 1, -N/2 + 1)$ ,  $n_1, n_2 = -M/2 + 1, \dots, M/2$ , of the adjacent frames/signal points  $(n_1, n_2)$  and  $(n_1, n_2 + 1)$ , from the same time direction  $n_1$ , but the adjacent time directions  $n_2$  and  $n_2 + 1$ , and
- Bordering S/SF points  $(n_1, M/2, N/2, N/2)$  and  $(n_1 + 1, -M/2 + 1, -N/2 + 1, -N/2 + 1)$ ,  $n_1 = -M/2 + 1, \dots, M/2$ , of the bordering frames/signal points  $(n_1, M/2)$  and  $(n_1 + 1, -M/2 + 1)$ , from the adjacent time directions  $n_1$  and  $n_1 + 1$ .

2. The *Completion* CLK of a frame/signal point  $(n_1, n_2)$  is overlapped in execution with the *Estimation* CLK of the bordering S/SF point  $(n_1, n_2, N/2, N/2)$  from the same

frame/signal point  $(n_1, n_2)$ , but also with the *SPEC execution* CLK of the adjacent bordering S/SF point  $(n_1, n_2 + 1, -N/2 + 1, -N/2 + 1)$  from the next frame/signal point  $(n_1, n_2 + 1)$ . In the same way, the *Completion* CLK of a bordering frame/signal point  $(n_1, M/2)$  is overlapped in execution with the *Estimation* CLK of the bordering S/SF point  $(n_1, M/2, N/2, N/2)$  from the same frame/signal point  $(n_1, M/2)$ , but also with the *SPEC execution* CLK of the adjacent bordering S/SF point  $(n_1 + 1, -M/2 + 1, -N/2 + 1, -N/2 + 1)$  from the next frame/signal point  $(n_1 + 1, -M/2 + 1)$ .

3. Finally, residual CLKs, as the conditional ones, are not necessarily performed in each S/SF point, so they cannot be included in overlapping in execution within the estimation.

LUT memory, Table 1, Fig. 6, manages the fully pipelined execution in each particular S/SF point. Besides the already described functions of the LUT signals *Completion* and *SHLorNo*, note that, in accordance with the 2D CTFWD definition, [17], [31], the *Termination* signal controls termination of the calculation corresponding to the maximum width  $L_m$  of the convolution window register block area, whereas the *SelSTFT\_1,2* signals determine address order of the STFT-to-CTFWD gateway's inputs, Fig. 2(a), [33].

Based on parameters from the *Configuration registers* and synchronized by considering conditions related to the CLK and *STFT\_Load/CTFWD\_Store* cycles, output signals of the *Ctrl of filtering & padding borders* block (set of modules that consist of variable length up-down binary counters and binary magnitude comparators) are generated to manage operations at the bordering frequency-frequency positions within the sliding

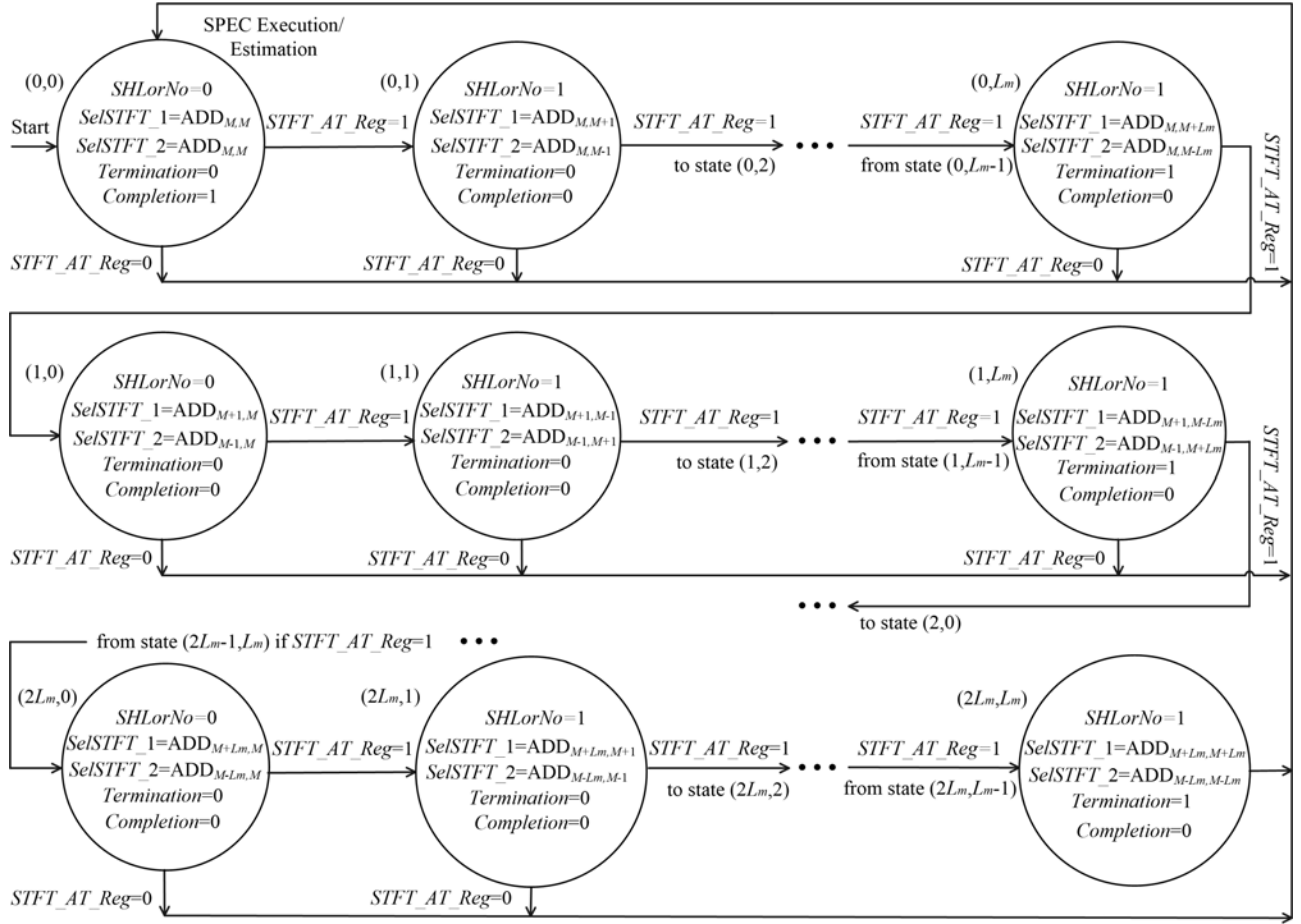


Fig. 6. The finite state machine for controlling the fully pipelined execution of the implementation shown in Fig. 1, Fig. 2. Inside the nodes, signals (from LUT) that manage the execution are asserted, whereas the other signals, as the signal adaptive ones, are calculated within the execution. The labels on arcs are conditions that are tested to determine which state is the next one (when the next state is unconditional, no labels is given). Return of the machine to the initial (0,0) state, provided by  $STFT\_AT\_Reg=0$  regardless the current state position, enables the beginning of the execution for the next S/SF point.

TABLE 1: LUT memory, presented in function of the address variable  $i$ ,  $i=0,1,\dots,2L_m$  and for an  $L_m$  value, determined by the 2D CTFWD definition, [17], [31], footnote 1.  $ADD_{M,M}$  denotes address of the central convolution window register block element,  $\ll$  denotes shift left logical operation,  $r=length(SelSTFT\_1)$ ,  $y=1$  for  $i=0$ ,  $y=0$  for  $i \neq 0$ , whereas the operator  $CEIL(i/2)$  rounds the value  $i/2$  to the nearest integer towards infinity.

LUT Address		Ctrl Signals Area			$SelSTFT\_1$	$SelSTFT\_2$
High	Low	$SHLorNo$	Termination	Completion		
$i$	0	0	0	$y$	$ADD_{M+CEIL(i/2),M} \ll r$	$ADD_{M-CEIL(i/2),M}$
$i$	1	1	0	0	$ADD_{M+CEIL(i/2),M+(-1)^i} \ll r$	$ADD_{M-CEIL(i/2),M-(-1)^i}$
$\vdots$	$\vdots$	1	0	0	$\vdots$	$\vdots$
$i$	$L_m$	1	1	0	$ADD_{M+CEIL(i/2),M+(-1)^i L_m} \ll r$	$ADD_{M-CEIL(i/2),M-(-1)^i L_m}$

of the convolution window and of the sliding matrix, Fig.3, as well as at the margins of the S/SF filtering process. By participation in the  $STFT\_AT\_Reg$  signal generation, Fig. 2(b), the  $Left\_Border\_1$  and  $Bottom\_Border\_1$  signals control padding the left and bottom borders with  $2L_m$  2D Spectrograms within the 2D CTFWD calculation. The  $Left\_Border\_2$  and  $Bottom\_Border\_2$  signals disable  $FRS_k$  setting in the bordering positions within the LF estimation, Fig. 1.

#### IV. TESTING AND VERIFICATION

To prove the proposed development, it is implemented, as

commonly done, [14]-[18], [35], [36], in an FPGA device. To verify its operation, the compilation and simulation are performed by considering the multicomponent noisy 2D signal consisting of the linear FM component and the pure sinusoid,

$$f(n_1T, n_2T) = \cos(20\pi(n_1T - 0.75)^2 + 22\pi(n_2T - 0.75)^2) + 0.5e^{j(-100\cos(\pi n_1T/2) + 100\cos(\pi n_2T/2))} \quad (4)$$

Signal (4) is corrupted by high white noise such that the input values of the signal-to-noise ratio (SNR) and the peak signal-to-noise ratio (PSNR) are  $SNR_m = 10 \times \log(P_s/P_n) = -0.9$  [dB] and



TABLE 2: Summarized resource utilizations (taken from compilation reports), maximum CLK rates, power consumptions (taken from the Altera’s PowerPlay Power Analyzer), and nominal area of the FPGA Cyclone III EP3C10E144C7 device used within the implementation of the proposed fully pipelined design and of the signal adaptive design [18], determined by  $N=64$ ,  $L_m=6$ ,  $L=4$ , and 2D STFT data length  $l=16$ . Dynamic (active) power consumption (the amount of power consumed when the device is operating) is calculated at CLK rate of 50MHz (CLK rate for which the real-time calculations are presented in Fig. 8), the static (standby) power consumption is calculated at 85°C junction temperature, while the I/O power represents the total power consumed by the device’s I/O pins.

FPGA implementation characteristics	Fully pipelined design	Signal adaptive design [18]
Chip Family	Cyclone III	Cyclone III
Recommended Device	EP3C10E144C7	EP3C10E144C7
Total Logic Elements	7,152/10,320 (69%)	7,194/10,320 (70%)
Combinational Functions	3,614/10,320 (35%)	3,924/10,320 (38%)
Dedicated Logic Registers	4,238/10,320 (41%)	4,708/10,320 (46%)
Total Pins	33/95 (35%)	33/95 (35%)
Total Virtual Pins	0	0
Total Memory Bits	13,298/423,936 (3%)	14,898/423,936 (4%)
Embedded Multiplier 9-bit elements	16/46 (35%)	16/46 (35%)
Total PLLs	0/2 (0%)	0/2 (0%)
Maximum CLK Rate (MHz)	200	200
Total (Dynamic+Static+I/O) Power Consumption (mW)	120.52 (45.27+53.89+21.36)	121.64 (46.41+53.94+21.29)
Nominal Device Area (mm <sup>2</sup> )	484 (22mm×22mm)	484 (22mm×22mm)

$PSNR_{in} = 10 \times \log(|\max(f(\vec{n}T))|^2 / MSE_{in}) = 3.83[\text{dB}]$ , respectively, where  $MSE_{in}$  is the mean squared error between the original signal and the input noisy one. Further, the structural similarity (SSIM) index, [37], also between the original signal and the input noisy one is  $SSIM_{in} = 0.1705$ . The noisy signal is observed within the range  $|n_1T, n_2T| \leq 0.75$ , where  $T = 1.5/M$  and  $M = 160$ . The 2D STFTs of the considered noisy signal, numerically calculated and written in the signed 16-bit fixed-point notation including 4-bit fraction, are imported (as presented in Fig. 1) to the input of the proposed design implemented in the FPGA EP3C10E144C7 device from the Cyclone III family, manufactured using the 65-nm technology and selected based on the optimal resource occupation. Within the numerical calculation, the Hanning lag-window  $w(\vec{m})$  is selected, as well as  $N = 64$ . As presented in Fig. 3, this signal occupies a wide range of frequencies (each signal component occupies the significant part of about 15% of whole frequency-frequency plane), while the minimal distance between different LFs of signal’s components can be compared with the range occupied by a particular component. Besides, the linear FM component of the considered signal is highly nonstationary one (normalized signal rates of this component in each particular of two signal dimensions are 0.89 and 0.97, respectively). Hence, signal (4) is selected as a very interesting and quite complex one for estimation, and because it can be used as an appropriate test for the resolution, selectivity and the estimation quality of the developed filter. Also, within the implementation,  $L_m = 6$ ,  $L = 4$ , and reference levels used within the 2D CTFWD calculation [17], [31], of  $R^2 = \max_{\vec{n}, \vec{k}} \{|STFT_x(\vec{n}, \vec{k})|^2\} / 100$ , and within the LF estimation of  $S^2 = \max_{\vec{n}, \vec{k}} \{CTFWD_x(\vec{n}, \vec{k})\} / 20$ , are selected.

Summarized resource utilizations for the FPGA device used within the implementation of the proposed fully pipelined design and of the design [18] are taken from the compilation reports and are given in Table 2, where implementations made on a same device additionally provide fair comparisons. As

the smallest units of logic, logic elements and their utilization are considered here. Namely, logic elements are the smallest units of logic in the Altera Corporation FPGA devices, such as the Cyclone III EP3C10E144C7 device selected here. These elements are compact and provide advanced features with efficient logic usage. In the Cyclone III family devices, each logic element has four inputs, a four-input LUT, a register, and output logic, where the four-input LUT is a function generator that can implement any function with four variables, [38].

Note that FPGA implementations of the considered designs consume low total power, Table 2, since the selected FPGA device in which designs are implemented, as the Cyclone III family device, provides the combination of high functionality, low power and low cost, [38]. However, in accordance with the improved resource utilization characteristics, the proposed design additionally provides the improved power consumption characteristics in comparison to the design [18], Table 2.

Considering maximum lengths of digital units used within the implementation, lengths of the output CumADD and the OutRegister of  $\text{CEIL}(\log_2((2^l - 1) \times N^2))$  are the critical points, where operator CEIL is described in the caption of Table 1. Besides, lengths of output registers of a multiplier, an adder, and a 2-input comparator, that determine the longest path of the implementation, are  $2 \times l$ ,  $\text{CEIL}(\log_2(2 \times (2^{2l} - 1)))$ , and  $2 \times l + 1$ , respectively, so the longest path doesn’t depend on any other implementation parameter ( $L$ ,  $L_m$ ,  $N$ ), but on the 2D STFT data length  $l$  only. Finally, within the execution, the maximum CLK rate of 200MHz is achieved, Table 2.

Filtering results are given in Fig. 7. High quality filtering can be noticed. The SNR improvement of  $SNR_{out} - SNR_{in} = 30.69[\text{dB}]$  ( $SNR_{out} = 29.79[\text{dB}]$ ) is achieved in the observed essentially 3-component signal case, Fig. 9, when the maximum SNR improvement of approximately  $10 \log(N \times N/3) = 31.3524[\text{dB}]$  can be theoretically expected (in an ideal situation and in the 3-component signal case, three LFs should be estimated, out of  $N \times N$  total S/SF points in which an LF is estimated per a frame/signal point). Note that these results are achieved despite the quite complex estimation of highly nonstationary components of the considered signal and the

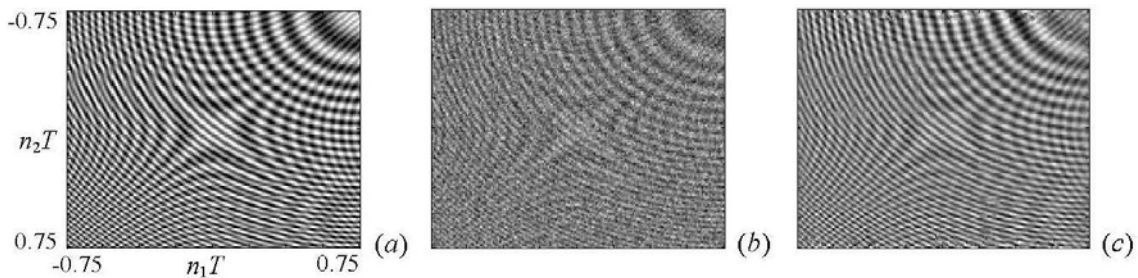


Fig. 7. (a) Original test signal (4), (b) Noisy signal (4), (c) Estimated signal obtained at the output of the proposed S/SF filter, implemented in the FPGA device.

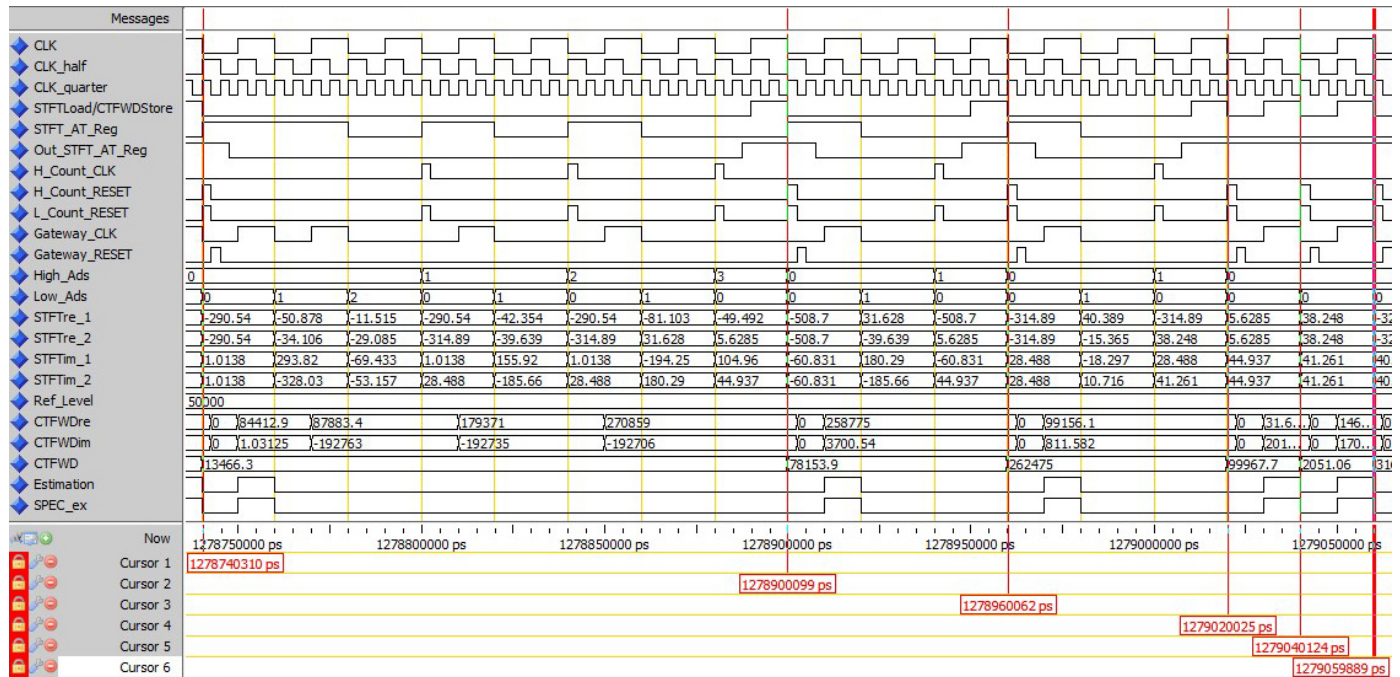


Fig. 8. Illustration of the real-time calculations.

To represent the complete executions per an S/SF point, S/SF points lying near the boundary of the 2D STFT auto-term's domain are considered (the S/SF points lying toward the center of an auto-term require great number of CLKs, Fig. 9, and the executions in these points cannot be completely represented). Note that instants at which the developed FPGA implementation is triggered completely correspond to the instants schematically represented and described in Fig. 4 (see instants at which High\_Ads, Low\_Ads, CTFWDre, and CTFWDim signals change their values, as well as instants at which CTFWDre and CTFWDim signals are reset and compare them with their schematic representation given in Fig. 4).

negative influence of the frequency discretization, [4], [18], footnote 2. The small difference between the achieved SNR improvement (of 30.69[dB]) and the theoretically maximal one (of 31.3524[dB]), caused by the noise influence and the frequency discretization, proves very high filtering efficiency. Besides, high quality resolution and selectivity achieved here correspond to the results achieved by the signal adaptive LF estimation-based solution from [18]. The PSNR improvement of  $PSNR_{out}-PSNR_{in}=33.04$ [dB] ( $PSNR_{out}=36.87$ [dB]) and the SSIM improvement of  $SSIM_{out}-SSIM_{in}=0.32$  ( $SSIM_{out}=0.4917$ ) are also achieved in the observed noisy signal case. These improvements can be considered as respectable ones, although in the aspect of the SSIM index, the observed pattern signal corrupted by the white noise in its full range, represents a very unfavorable example for the similarity evaluation.

Main contributions of the proposed fully pipelined design are illustrated in Fig. 8 and in Fig. 9. Signals *Estimation* and *SPEC\_ex*, given in Fig. 8, respectively present *Estimation* and *SPEC execution* CLKs performed in different S/SF points. As seen, these signals are identical, which means that *Estimation*

and *SPEC execution* CLKs, performed in the adjacent S/SF points, are overlapped in execution. Besides, calculations performed in S/SF points lying outside the 2D STFT auto-term's domains (such as the calculations performed between the cursors 4 and 5 positions, and between the cursors 5 and 6 positions in Fig. 8) take only one CLK, despite two unconditional CLKs (*SPEC execution* and *Estimation* ones) that, as already noted, have to be performed within the execution in each S/SF point. This can only be explained by the fact that, in these points, 2D CTFWDs are obtained on every CLK cycle, while the other unconditional-*Estimation*-CLK, also performed in each of these points, is overlapped in execution with the 2D CTFWD calculation performed in the next S/SF point. Distribution of CLKs, performed by the proposed design in different S/SF points for a particular signal point (0.25,0.25) and within the noisy signal (4) estimation, is given by the gray-scale graph in Fig. 9 and is compared with the signal adaptive design [18]. As can be noted, the fully pipelined design comparatively improves execution time by a CLK per each S/SF point performed within the execution.

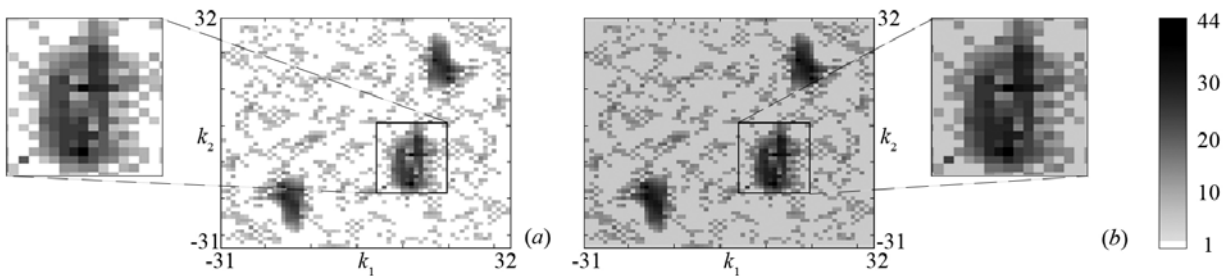


Fig. 9. CLKs distribution per an S/SF point in the signal point (0.25,0.25). (a) Proposed fully pipelined design, (b) Signal adaptive partly pipelined design [18].

## V. COMPARATIVE ANALYSIS AND CONCLUSIONS

Properties of the proposed fully pipelined design follow the comparative analysis presented in Tables 3–5. The proposed design is firstly compared with the corresponding signal adaptive, but only partly pipelined LF estimation-based S/SF filter implementation from [18], Table 3. This comparison has the general significance, especially in the case of the proposed design superiority, since the design [18] optimizes, in all the aspects considered here, all other possible LF estimation-based S/SF filtering implementations (the classical single-clock-cycle one (SCI) with a fixed CLK, the classical MCI one with a fixed number of CLKs, and the hybrid one<sup>4</sup>), as explicitly proven in [18]. After that, comparisons with the state-of-the-art linear and non-linear S/SF filtering solutions are performed and are given in Tables 4–5.

As seen, only the design [18] is competitive to the design proposed in this paper, but only in the hardware complexity aspect. In all the other considered aspects and regarding all the considered filtering solutions, the proposed solution is superior. Firstly, note that the proposed design, as the essential MCI one, includes a fixed number of functional units, and, hence, significantly improves corresponding requirements of the SCI state-of-the-art S/SF filtering solutions, considered in Table 4. Namely, each functional unit can be used once per a particular CLK, but MCIs, such as one proposed here and the one from [18], take multiple (only necessary number of) CLKs per an S/SF point and, therefore, allow a functional unit to be used more than once within the execution in an S/SF point, as long as it is used in different CLKs. Accordingly, repetition of functional units is not required in the MCIs and their hardware complexities do not depend either on the windowed 2D signal duration  $N \times N$ , or on the maximum 2D convolution window width  $(2L_m+1) \times (2L_m+1)$ . Besides, these systems are always suitable for on-a-chip implementation and for signals of

<sup>4</sup> Within the execution in an S/SF point, the classical SCI takes only one CLK (within its first half, the 2D CTFWD calculation corresponding to the maximum convolution window width is performed, [39], and within its second half, the LF is estimated), whereas the classical MCI takes higher, but fixed number of  $2L_m^2 + 2L_m + 3$  CLKs, [16], where the LF estimation is performed in the last CLK (after the multiple-clock-cycle 2D CTFWD calculation corresponding to the maximum convolution window width, [16]). Considering these two approaches, the SCI one optimizes execution time, whereas the MCI one minimizes the CLK time and the hardware complexity, [16]. As a solution with compromise performances, the hybrid implementation approach, [18], [40], is the MCI one, but within its CLK time, the single-clock-cycle 2D CTFWD calculation, [39], corresponding to the fixed convolution window width of  $(2L_h+1) \times (2L_h+1)$ ,  $1 < L_h \leq L_m$ , is performed, [18], instead of the calculation corresponding to the convolution window width of  $(1 \times 1)$ , as performed here, in [17], [18], and in the classical MCI case, [16].

arbitrary duration (signals of small or high duration  $N \times N$ ).

Opposed to the MCIs, the SCIs take only one CLK within the execution in an S/SF point, so these systems require repeating of functional units if they need to be used more than once. Therefore, depending on the filter definition, their hardware complexity depends either on the windowed 2D signal duration  $N \times N$  (Table 4) or on the 2D convolution window widths  $(2L_m+1) \times (2L_m+1)$  or  $(2L_h+1) \times (2L_h+1)$ , [18], [39], footnote 4. Accordingly, these systems do not satisfy any of the noted characteristics (of the MCIs) following from the hardware complexity independence on  $N$ ,  $L_m$  and  $L_h$ .

Note also that, because of lower LUT memory capacity, the proposed design improves the total memory capacity of the signal adaptive design [18] for one location, Tables 3, 4. This can be explained by the fact that the control signals saved in the 0th LUT location of the proposed design simultaneously manage the executions within the *SPEC execution*, *Estimation*, and *Completion* CLKs. In this way, since the LUT memory manages the execution in each particular S/SF point, the missing location implies overlapping in execution corresponding to one CLK per an S/SF point performed within the execution, but also corresponding to one additional CLK per a frame/signal point, that finally results in saving of  $M \times M \times N \times N$  CLKs (in comparison to [18]) within the filtering. Namely, within the S/SF analysis-based filtering, a great number of  $M \times M \times N \times N$  S/SF points have to be included in execution, Fig. 5. Hence, the improvement in execution time, expressed by the number of saved CLKs and achieved by the proposed design in comparison to the design [18], corresponds to the total number of S/SF points included in execution (i.e., in the example observed in this paper, this improvement corresponds to  $160 \times 160 \times 64 \times 64 = 104857600$  saved CLKs).

As already noted, regarding the execution time and the calculation complexity aspects, the proposed design shows superior performances, Tables 3–4. However, the improvement reached in execution time is the essential one. To prove this, the comparisons of the proposed design with the corresponding classical SCI one, [39], [18], and the signal adaptive MCI one from [18] are highlighted here. Namely, the execution time represents the main drawback of the classical MCI and hybrid designs in comparison to the classical SCI one, [16], while design [18], as the signal adaptive and a partly pipelined one (within its implementation, only one CLK–*Completion* one—between the adjacent frames/signal points, is pipelined, but the execution performing in a frame/signal point is non-pipelined), improves both the classical implementation approaches and the hybrid one in the execution time aspect. The following conclusions can be derived:

1. SCI of the LF estimation-based S/SF filter (1)-(2) takes

TABLE 3: Proposed fully pipelined versus the signal adaptive LF estimation-based MCI S/SF filter design [18] from the hardware complexity and the execution time aspects. Essentially, all the other possible LF estimation-based S/SF filter implementations (the classical SCI and MCI ones, as well as the hybrid one) are also included into consideration, since the design [18] optimizes their time and hardware complexity performances, as proven in [18]. Add, Mult, ShLeft denote adders, multipliers, and shift left registers,  $\lambda$  ( $2 \leq \lambda \leq 2L_m^2 + 3L_m + 2$ ) is the average number of CLKs taken per an S/SF point within the non-pipelined execution in a signal point [18] ( $\lambda=7.2094$  in the example given in this paper), and  $T_c/2=T_m+T_a+T_{comp}$  is the CLK time of the proposed design and the design [18].

Design	# functional units			# of memory locations	Sampling rate	Execution time per an S/SF point
	Add	Mult	ShLeft			
Signal adaptive [18]	6	6	2	$2N^2 + 4NL_m + 2L_m^2 + 7L_m + 3NL + 3L + 20$	$1/(2 \times T_c)$	$\lambda \times T_c$
Fully pipelined	6	6	2	$2N^2 + 4NL_m + 2L_m^2 + 7L_m + 3NL + 3L + 19$	$1/T_c$	$(\lambda - 1) \times T_c$

only one CLK per an S/SF point (see footnote 4). Hence, its execution time per an S/SF point corresponds to a CLK time that is  $2(2T_m + (2L_m^2 + 2L_m + 2)T_a + T_s)$ , [18], [39]. Opposed to the SCI design, the design proposed in this paper provides fully pipelined execution, takes multiple, but signal adaptive number of CLKs per an S/SF point, and, therefore, its execution time depends on the estimated signal shape and the noise distribution, as taken into account by the factor  $\lambda$  introduction within the execution time calculation (Table 3). Accordingly, regarding the execution time aspect and depending on the  $\lambda$  value, the proposed design can significantly surpass the corresponding SCI one, removing the main drawback of the classical MCIs in comparison to the corresponding SCIs. For example, in the analyzed noisy signal (4) case, when  $\lambda=7.2094$ , the proposed design with  $L_m=6$  improves the execution time in comparison to the corresponding classical SCI design for  $T_s, T_{comp} \ll T_m < 19.955 \times T_a$ .

2. Opposed to the single adaptive, but only partly pipelined design from [18], the proposed one provides overlapping in execution corresponding to one CLK within the execution in each particular S/SF point. In this way, if  $\lambda$  denotes the average number of CLKs taken per S/SF point within the non-pipelined execution in a frame/signal point (the execution corresponding to the signal adaptive design from [18]), the proposed fully pipelined design takes  $(\lambda-1)$  CLKs within the execution in an S/SF point, or, comparatively, improves execution time by a CLK per S/SF point performed within the estimation. Relatively, this means improvements of  $(100/\lambda)\%$ , per output signal point (13.87% in the example analyzed within the verification, when  $\lambda=7.2094$ ), and of 50%, in terms of S/SF points lying outside the 2D STFT auto-terms' domains (in these points 2 CLKs are performed within the non-pipelined execution in a frame/signal point).

Throughput, defined as the total amount of work done in a given time, is, in the case of considered systems, the reciprocal of the execution time taken per a frame/signal point (i.e.,  $1/(N \times N \times (\lambda-1) \times T_c)$  in the proposed system case, Table 3). Accordingly, the proposed fully pipelined design improves the throughput of the design [18] and this improvement (of  $(100/\lambda)\%$ ) corresponds to the improvement achieved in execution time, where “improve throughput” means “increase throughput”, while “improve execution time” means “decrease execution time”. In addition, since CLK times of the proposed design and the design from [18] are determined by the

multiplication and addition times ( $T_c/2=T_m+T_a+T_{comp}$ , where  $T_{comp} \ll T_m, T_a$ ), execution times of these systems taken per a frame/signal point are respectively determined by  $O(N^2 \times 4(\lambda-1))$  and by  $O(N^2 \times 4\lambda)$  operations performed per an output signal sample (for  $N=64$ ,  $\lambda=7.2094$ , the considered execution times are respectively determined by  $O(99.35K)$  and by  $O(115.35K)$  operations performed per an output signal sample, where  $1K=2^{10}$ ). Contrarily, the other considered filters in Tables 4–5, as the SCI solutions, require execution times taken per a frame/signal point determined by the total number of operations performed per an output signal sample. Based on these observations and on the calculation complexities of the SCI solutions given in Tables 4–5, it can easily be concluded that the proposed fully pipelined design simultaneously improves execution times and throughputs of all the considered state-of-the-art filters (for  $N=64$ ,  $\lambda=7.2094$ ,  $\xi=2$ , the proposed design improves execution time and throughput of the classical (single-window) Gabor filter, as the best one among the other considered solutions, for 11.59%).

Further, the signal adaptive designs (the proposed one and the design from [18]) import input data with the fixed rate determined by the minimal execution time per an S/SF point. Consequently, in these systems, the sampling rates of the input analog signal are also determined by the minimal execution time per an S/SF point. Since the proposed design and design from [18] take the minimal execution times of a CLK and of two CLKs per an S/SF point, respectively, Fig. 9, the proposed design imports input data on each CLK cycle, or twice as fast as the partly pipelined design [18], and simultaneously provides discretization of the estimated input analog signal with the fixed, but twice faster sampling rate. Finally, in comparison to the design [18], the proposed fully pipelined one substantially simplifies the control managed by the LUT memory. Accordingly, within the implementation on the same FPGA EP3C10E144C7 device, the proposed design additionally reduces (in comparison to the hardware complexity consideration given in Table 3) the device's resource utilization, as well as the total power consumption, Table 2.

Simultaneously, the S/SF filter proposed here is compared (Tables 4–5) to the online designs of the state-of-the-art non-stationary filtering solutions [1]–[3], [19], [21]–[23], extended to their 2D forms. To provide full and fair comparisons regarding the critical design performances, the solutions based on the S/SF analysis and the S/SF analysis tools are included in consideration. The state-of-the-art extended and improved nonstationary filtering solutions, including the multiwindow STFT and Gabor filters, several forms of the halfband Weyl



TABLE 4: Proposed fully pipelined versus the state-of-the-art online nonstationary filters [1]-[3], [19], [21]-[23], extended to their 2D forms, from the hardware and the calculation complexity aspects. Essentially, the input halfband, output halfband, and halfband Weyl filters are also included here (minimum energy Weyl filter minimizes their complexities, [1]-[3]), as well as the classical (single-window) Gabor and STFT filters (follow from their multiwindow forms for  $M_w=1$ ) and the direct matched filter (the improved matched filter from [21] minimizes its complexity). Operations that determine filters' execution times are included, except shift logical ones (the other operations require much longer time for their executions). Parameters used in the corresponding filters' designs are  $m=\log_2(N)$ ,  $\xi \geq 1$ ,  $M_w \geq 1$ , and  $5 \leq p \leq 7$ , [1]-[3], [19], while a constant  $c$  depends on the specific FFT algorithm used in the matched filters, [21], and  $1 \leq \Delta \leq 2N-1$ , [22].

Filter type	Hardware complexity		# of operations per output signal sample
	# of functional units	# of memory locations	
Zadeh	$N^2+N^3(m+1)$	$N^2+N^4/2$	$O(N^2+N^3(m+1))$
Multiwindow STFT	$M_w(3N^2+2N^3(m+1))$	$2M_wN^2$	$O(M_w(3N^2+2N^3(m+1)))$
Multiwindow Gabor	$M_w\xi(3N+2N^2(m+1))$	$2M_wN^2$	$O(M_w\xi(3N+2N^2(m+1)))$
Minimum-energy Weyl	$N^2+N^3(m+1)$	$2N^2+N^4/2$	$O(N^2+N^3(m+1))$
Approximate halfband Weyl	$N^2/4+mN^3/8$	$N^2+N^4/8$	$O(N^2/4+mN^3/8)$
Projection	$N^2(3N^2p+3Np^2+p^3+N^2+mN/8)$	$N^2(2N^2+2p^2+Np+1)$	$O(N^2(3N^2p+3Np^2+p^3+N^2+mN/8))$
Matched [21]	$cmN^4/2$	$4N^4$	$O(cmN^4/2)$
Matched approximated by the sum-of-spectrograms method [22], [23]	$c\Delta mN^4/8$	$\Delta N^4$	$O(c\Delta mN^4/8)$
Matched [21] augmented by S/SF weighting function	$3cmN^4/8$	$4N^4$	$O(3cmN^4/8)$
Signal adaptive [18]	14	$2N^2+(4N+2L_m+7)L_m+3NL+3L+20$	$O(N^2 \times 5(\lambda-1)+2N^2)$
Fully pipelined	14	$2N^2+(4N+2L_m+7)L_m+3NL+3L+19$	$O(N^2 \times 5(\lambda-1))$

TABLE 5: Real implementation results of state-of-the-art filters considered in Table 4, obtained for parameters' values  $N=64$ ,  $L_m=6$ ,  $L=4$ ,  $M_w=10$ ,  $\lambda=7.2094$ ,  $p=5$ ,  $c=1$ ,  $\xi=2$  and  $\Delta=5$  (these values are suggested in [1]-[3], [19]-[23], [41]). Observed filter types are: 1- Zadeh, 2- multiwindow STFT, 3- multiwindow Gabor, 4- single-window STFT, 5- single-window Gabor, 6- minimum energy Weyl, 7- approximate halfband Weyl, 8- projection, 9- matched, [21], 10- matched approximated by the sum-of-spectrograms method [22], [23], 11- matched [21] augmented by S/SF weighting function, 12- signal adaptive [18], and 13- the proposed fully pipelined one, whereas  $1K=2^{10}$ ,  $1M=2^{20}$ .

Filter type	Hardware complexity		# operations per output sig. sample
	# funct. units	# mem. locations	
1	1.754M	8.004M	$O(1.754M)$
2	35.117M	80K	$O(35.117M)$
3	1.097M	80K	$O(1.097M)$
4	3.512M	8K	$O(3.512M)$
5	112.375K	8K	$O(112.375K)$
6	1.754M	8.008M	$O(1.754M)$
7	193K	2.004M	$O(193K)$
8	275.426M	33.449M	$O(275.426M)$
9	48M	64M	$O(48M)$
10	60M	320M	$O(60M)$
11	36M	64M	$O(36M)$
12	14	10.393K	$O(132.188K)$
13	14	10.392K	$O(124.188K)$

filter, the minimum energy Weyl filter, the recently proposed and improved matched filters, and the signal adaptive filter from [18], are taken into consideration. In addition, the classical nonstationary filtering solutions are also included in consideration, but implicitly (as the special cases of their extended versions), as noted in the caption of Table 4.

As seen, the proposed design optimizes hardware and

calculation complexity of the considered filters, whereas only the classical (single-window) STFT and Gabor filters, following from their multiwindow forms for  $M_w=1$ , improve the total memory capacity of the proposed solution. In detail, the total memory capacities of the classical STFT and Gabor filters (of  $2N^2$  locations) minimize the total memory capacities of the other considered solutions (the proposed design capacity of order of  $2N^2+(4L_m+3L) \times N$  locations, the multiwindow STFT and Gabor filters' capacities of  $2M_wN^2$  locations, where  $M_w>1$ , and the other capacities of order of  $N^4$  locations). However, among all the considered solutions, only the filter developed here and the signal adaptive one from [18], as the MCI ones, require both the fixed number of functional units and the fairly small memory capacities. Thus, only they are suitable for signals of arbitrary duration, as well as for real time and system-on-a-chip implementation. Contrarily, the other considered filters, as the SCI ones, have enormous hardware complexities, Table 5, so their implementations require chips of significantly greater dimensions and are not always possible (for higher  $N \times N$ ), power consumed by these systems (when possible) is substantially increased, as well as their cost, [16]. Hence, considering state-of-the-art filters given in Tables 4-5, the filter developed here and the filter [18] optimize critical design performances related to the hardware complexity, such as the chip dimensions, power consumption, and cost. Finally, as already noted, regarding these performances, the developed filter additionally improves the filter from [18], Table 2.

## REFERENCES

- [1] G. Matz, F. Hlawatsch: "Linear time-frequency filters: Online algorithms and applications," in *Appl. Time-Frequency Signal Process.* (A. Papandreou-Suppappola, ed.), CRC Press, 2002, pp. 205-271.

- [2] G. Matz, F. Hlawatsch: "Linear time-frequency filters," in *Time-Freq. Sig. Analysis & Process.* (B. Boashash, ed.), Elsevier, 2016, pp. 638-645.
- [3] F. Hlawatsch, G. Matz, H. Kirchauer, W. Kozek: "Time-frequency formulation, design and implementation of time-varying optimal filters for signal estimation," *IEEE Trans. Sig. Process.*, vol. 48, no. 5, May 2000, pp. 1417-1432.
- [4] L.J. Stanković: "Space/spatial-frequency analysis based filtering," *IEEE Trans. on Signal Process.*, vol. 48, no. 8, Aug. 2000, pp. 2343-2352.
- [5] G.F. Boudreaux-Bartels: "Time-varying signal processing using Wigner distribution synthesis techniques," in *The Wigner distribution – theory and applications in sig. process.* (W. Mecklenbrauker, F. Hlawatsch, eds.), Elsevier, 1997, pp. 269-317.
- [6] A. Beghdadi, R. Iordache: "Image quality assessment using the joint spatial/spatial-frequency representation," *EURASIP J Applied Sig. Process.*, vol. 2006, 2006, pp. 1–8.
- [7] I. Orović, S. Stanković: "L-statistics based space/spatial-frequency filtering of 2D signals in heavy tailed noise," *Sig. Process.*, vol. 96, March 2014, pp. 190-202.
- [8] S. Wang, Y. Xia, Q. Liu, J. Luo, Y. Zhu, D.D. Feng: "Gabor feature based nonlocal means filter for textured image denoising," *J Visual Comm. & Image Represent.*, vol. 23, no. 7, Oct. 2012, pp. 1008-1018.
- [9] S. Stanković, I. Orović, M. Chabert, B. Mobasserri: "Image watermarking based on the space/spatial-frequency analysis and Hermite functions expansion," *J Electronic Imaging*, vol. 22, no. 1, 2013.
- [10] J. Hormigo, G. Cristóbal: "Image segmentation using the Wigner-Ville distribution," *Adv. in Imaging & Electron. Physics*, vol. 131, Dec. 2004, pp. 65-80.
- [11] R. Redondo, S. Fischer, F. Sroubek, G. Cristóbal: "A 2D Wigner Distribution-based multisize windows technique for image fusion," *J. Visual Comm. & Image Represent.*, vol. 19, no. 1, Jan. 2008, pp. 12-19.
- [12] J. Harmigo, G. Cristóbal: "High resolution spectral analysis of images using the pseudo-Wigner distribution," *IEEE Trans. Sig. Process.*, vol. 46, no. 6, June 1998, pp. 1757-1763.
- [13] W.D. Furlan, C. Soriano, G. Saavedra: "Opto-digital tomographic reconstruction of the Wigner distribution function of complex fields," *Applied Optics*, vol. 47, no. 22, Aug. 2008, pp. E63-E67.
- [14] J. Liu-Jimenez, R. Sanchez-Reillo, B. Fernandez-Saavedra: "Iris Biometrics for Embedded Systems," *IEEE Trans. VLSI Systems*, vol. 19, no. 2, Feb. 2011, pp. 274-282.
- [15] M. Fons, F. Fons, E. Canto, M. Lopez: "FPGA-based Personal Authentication Using Fingerprints," *J Signal Process. Systems*, 66 (2012), pp. 153-189.
- [16] V.N. Ivanović, R. Stojanović: "An efficient hardware design of the flexible 2-D system for space/spatial-frequency analysis," *IEEE Trans. Signal Process.*, vol. 55, no. 6, June 2007, pp. 3116-3126.
- [17] V.N. Ivanović, S. Jovanovski: "Signal adaptive system for space/spatial-frequency analysis," *EURASIP J Adv. in Signal Process.*, 2009, pp. 1-15.
- [18] V.N. Ivanović, N. Radović: "Signal adaptive hardware implementation of a system for highly nonstationary two-dimensional FM signal estimation," *AEU – Int. J Electron. & Comm.*, vol. 69, no. 12, Dec. 2015, pp. 1854–1867.
- [19] G. Matz, F. Hlawatsch: "Time-frequency projection filters: Online implementation, subspace tracking, and application to interference suppression," *Proc. ICASSP 2002*, Orlando, FL, 2002, pp. 1213-1216.
- [20] B. Boashash: "Signal denoising by time-frequency peak filtering (TFPF)," in *Time-Freq. Sig. Analysis & Process.* (B. Boashash, ed.), Elsevier, 2016, pp. 663-670.
- [21] J.M. O'Toole, M. Mesbah, B. Boashash: "Accurate and efficient implementation of the time-frequency matched filter," *IET Sig. Process.*, vol. 4, no. 4, 2010, pp. 428-437.
- [22] A.M. Sayeed: "Optimal time-frequency detectors," in *Time-Freq. Sig. Analysis & Process.* (B. Boashash, ed.), Elsevier, 2016, pp. 694-701.
- [23] B. Boashash, G. Azemi: "A general approach to time-frequency-matched filtering," in *Time-Freq. Sig. Analysis & Process.* (B. Boashash, ed.), Elsevier, 2016, pp. 724-732.
- [24] G.W. Mabey, T. Bose, M.-Q. Chen: "Stability of shift-varying 2-D state-space digital filters," *IEEE Trans. Circ. Syst. I*, vol. 59, no. 7, July 2012, pp. 1431-1444.
- [25] S. Jovanovski, V.N. Ivanović: "Signal adaptive pipelined hardware design of time-varying optimal filter for highly nonstationary FM signal estimation," *J Signal Process. Systems*, 62(3), 2011, pp. 287-300.
- [26] V.N. Ivanović, N. Radović, S. Jovanovski: "Real-time design of space/spatial-frequency optimal filter," *El. Lett.*, vol. 46, no. 25, Dec. 2010, pp. 1696-1697.
- [27] D. Wu, D. Zhu, M. Shen, Z. Zhu: "Time-varying space-time autoregressive filtering algorithm for space-time adaptive processing," *IET Radar, Sonar & Navigation*, vol. 6, no. 4, 2012, pp. 213-221.
- [28] A. Papoulis, *Signal Analysis*, McGraw-Hill, New York, USA, 1997.
- [29] V.N. Ivanović, M. Daković, L.J. Stanković: "Performances of quadratic time-frequency distributions as instantaneous frequency estimators," *IEEE Trans. on Sig. Process.*, vol. 51, no. 1, Jan. 2003, pp. 77-89.
- [30] I. Djurović, L.J. Stanković, J.F. Böhme: "Estimates of the Wigner distribution in Gaussian noise environment," *AEU – Int. J Electron. & Comm.*, vol. 56, no. 5, 2002, pp. 337-340.
- [31] V.N. Ivanović, S. Jovanovski: "Signal adaptive method for improved space/spatial-frequency representation," *El. Lett.*, vol. 45, no. 19, Sept. 2009, pp. 1003–1005.
- [32] D.A. Patterson, J.L. Hennessy, *Computer organization & Design: The hardware/software interface*, Elsevier, San Francisco, CA, USA, 2005.
- [33] V.N. Ivanović, S. Jovanovski: "A signal adaptive system for time-frequency analysis," *El. Lett.*, vol. 44, no. 21, Oct. 2008, pp. 1279-1280.
- [34] A. Joy, V.K. Chakka: "Fast array multichannel 2D-RLS based OFDM channel estimator," *Circuits, Systems and Signal Process.*, vol. 32, no. 3, 2013, pp. 1419-1432.
- [35] A. Gabiger-Rose, M. Kube, R. Weigel, R. Rose: "An FPGA-based fully synchronized design of a bilateral filter for real-time image denoising," *IEEE Trans. Indus. Electron.*, vol. 61, no. 8, Aug. 2014, pp. 4093-4104.
- [36] A. Rosado-Muñoz, M. Bataller-Mompeán, E. Soria-Olivas, C. Scarante, J.F. Guerrero-Martínez: "FPGA Implementation of an Adaptive Filter Robust to Impulsive Noise: Two Approaches," *IEEE Trans. Industrial Electron.*, vol. 58, no. 3, March 2011, pp. 860-870.
- [37] Zhou Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli: "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, April 2004, pp. 600-612.
- [38] Altera, "Cyclone III Device Handbook," San Jose, CA 95134, vol. 1, Version 2.0, Aug. 2012. [Online]. Available: [https://www.altera.com/en\\_US/pdfs/literature/hb/cyc3/cyclone3\\_handbook.pdf](https://www.altera.com/en_US/pdfs/literature/hb/cyc3/cyclone3_handbook.pdf)
- [39] S. Stanković, I. Djurović, V. Vuković: "System architecture for space-frequency image analysis," *El. Lett.*, vol. 34, no. 23, Nov. 1998, pp. 2224-2225.
- [40] V.N. Ivanović, R. Stojanović, L.J. Stanković, "Multiple clock cycle architecture for the VLSI design of a system for time-frequency analysis," *EURASIP J App. Sig. Process., Spec. Issue: Design Methods for DSP Syst.*, vol. 2006, pp. 1-18.
- [41] G. Cunningham, W. Williams: "Fast implementations of generalized discrete time-frequency distributions," *IEEE Trans. Sig. Process.*, vol. 42, no. 6, June 1994, pp. 1496–1508.



**Veselin N. Ivanović** (A'02–M'08–SM'10) was born in Cetinje, Montenegro (1970). In 1993, 1996, and 2001, he received the B.S., M.S., and Ph.D. degrees in Electrical Engineering, all from the University of Montenegro. In 2001, he received the Siemens Award for scientific achievements in his Ph.D. research.

Since 1994, he has been at the EE Department, University of Montenegro, where he is a Full Professor (since 2012). His current research interests are in the areas of Signal Processing, Time- and Space/Spatial-Frequency Signal Analysis, and Design and Implementation of Signal Processing Systems.

Prof Ivanović was a member of a research group that received a research award grant for 2001 to 2003 from the Volkswagen Foundation, Federal Republic of Germany. He was a member of committees of some international conferences and is a reviewer in several world leading journals, including IEEE Trans. on Signal and on Image Processing, and Signal Processing (Elsevier). He is a Senior member of the IEEE and an EURASIP member.



**Nevena R. Brnović** was born in Podgorica, Montenegro (1984). She received BSc, MSc, and Ph.D. degrees in electrical engineering, all from the University of Montenegro in 2005, 2007, and 2014, as well as the Siemens Award for best student in 2006.

She is a Teaching Assistant at the EE Department, University of Montenegro. Her research interests are in the areas of signal processing, time-/space/spatial-frequency analysis, implementation of signal processing systems, and hardware/software co-design.