SIGNAL ADAPTIVE HARDWARE DESIGN OF A SYSTEM FOR HIGHLY NONSTATIONARY FM SIGNAL ESTIMATION

Veselin N. Ivanović, Srdjan Jovanovski

Dept. of Electrical Engineering, University of Montenegro
81000 Podgorica, MONTENEGRO
phones: + (382) 67 331 866, + (382) 69 453 995, fax: + (382) 245 839, email: *very@ac.me*, *srdjaj@t-com.me*

ABSTRACT

Multiple-clock-cycle implementation (MCI) of a signal adaptive optimal nonstationary filtering system is developed. The proposed design is based on the real-time results of time-frequency (TF) analysis, on the correspondence of filter's region of support (FRS) to the signal's instantaneous frequency (IF), and on the real-time IF estimation. It permits multiple local FRS detection in the observed time-instant, resulting in the efficient filtering of multicomponent FM signals. It takes a variable number of CLKs–the only necessary ones regarding the highest quality of IF estimation–in different TF points within the execution. The design also optimizes critical performances related to the hardware complexity, making it a suitable system for implementation on an integrated chip.

1. INTRODUCTION

Efficient estimation of nonstationary signals requires a time-varying (TV) approach that may then benefit from the TF analysis results. Linear TF filters, applications and quite complex algorithms for their implementation have already been studied, [1]. Nonlinear filters, related to the Wigner distribution (WD), have also been studied, [2, 3], as well as their implementation approaches, [3, 4]. However, being quite complex, [3, 4], and unsuitable in the multicomponent signals case, [4], these approaches are unsuitable for real-time implementation. To overcome complexity of linear and nonlinear solutions, but to simultaneously increase their time requirements, the MCI of the WD-based optimal nonstationary filter has been proposed in [5]. In this paper we improve MCI design from [5] by introducing signal adaptive solution that retains desirable characteristics of the non-adaptive solution [5] (related to the hardware complexity), but also significantly improves execution time, even in comparison to the single-cycle solutions, [1, 4]. The execution time improvement is achieved by taking a variable number of only necessary CLKs (regarding the highest IF estimation quality) in different TF points within the execution.

2. THEORY BACKGROUND

TV filtering definition, based on Weyl correspondence, [1–3], that overcomes distortion of the filtered FM signal is, [3]:

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

$L_H(n,k)$ is the FRS, $STFT_x(n,k)=DFT_m\{w(m)x(n+m)\}$ is the short-time FT (STFT) of the q-component noisy signal $x(n) = \sum_{i=1}^{q} f_i(n) + \varepsilon(n)$, $w(m)$ is the real-valued lag window and N is the signal duration.

Considering a single realization of the FM signals $f_i(n)$, i=1,…,q, highly concentrated in TF plane and exposed to a widely spread white noise, the FRS of the optimal TV filter corresponds to the combination of the local IFs of signals $f_i(n)$, [2, 3]. Therefore, the filtering problem can be reduced to the local IF estimation in a noisy environment. Since the cross-terms-free WD (CTFWD), [6], produces the best IF estimation characteristics in the case of highly nonstationary signals and the TF analysis framework, [7], the local IF estimation is performed here by determining frequency points where CTFWD of the noisy signal has local maxima, [7],

$$IF_i(n) = \arg[\max_{k\in Q_{k_i}} CTFWD_x(n,k)]. \qquad (2)$$

Q_{k_i} is the basic interval around $f_i(n)$, whose IF is $IF_i(n)$.

TV filter for highly nonstationary FM signals estimation is developed here following the real-time IF estimation algorithm, proposed in [8], and using the available MCI CTFWD real-time design, [6], to provide the improved noisy signal $x(n)$ TF representation (TFR). Besides, the CTFWD, [6],

$$CTFWD_x(n,k) = \left|STFT_x(n,k)\right|^2 + 2\sum_{i=1}^{L(n,k)} \mathrm{Re}\{STFT_x(n,k+i) STFT_x^*(n,k-i)\} \qquad (3)$$

requires the STFTs, used also in the definition (1). $L(n,k) \le L_m$ is the signal adaptive width of the rectangular convolution window, introduced to limit convolution of the STFTs[1] and then to enable production of the pure cross-terms-free TFR,

[1] To preserve WD auto-terms, the convolution of the STFTs in (3) must be performed, for each point (n,k), until $STFT_x(n,k\pm i)=0$, or, practically (noisy signals case), until $|STFT_x(n,k\pm i)|^2<S^2$, is detected (S^2 is a predefined reference level, determined as a few percent of the spectrogram's (SPEC's) maximal value, [6]). It means that boundaries of each STFT auto-term's domain coincide with the detection of $|STFT_x(n,k\pm i)|^2<S^2$ around corresponding signal component ($|STFT_x(n,k\pm i)|^2 \ge S^2$, for i=0,1,…,$L(n,k)$, in each point (n,k) from STFT auto-terms' domains, whereas $|STFT_x(n,k\pm i)|^2<S^2$ $\forall i$ otherwise). Then, $L(n,k)$ takes variable values in different TF points: zero ($L(n,k)=0$) outside STFT auto-terms' domains and at their boundaries, the higher one inside these domains, and the maximum one (L_m) only in the central points of the widest domain(s).

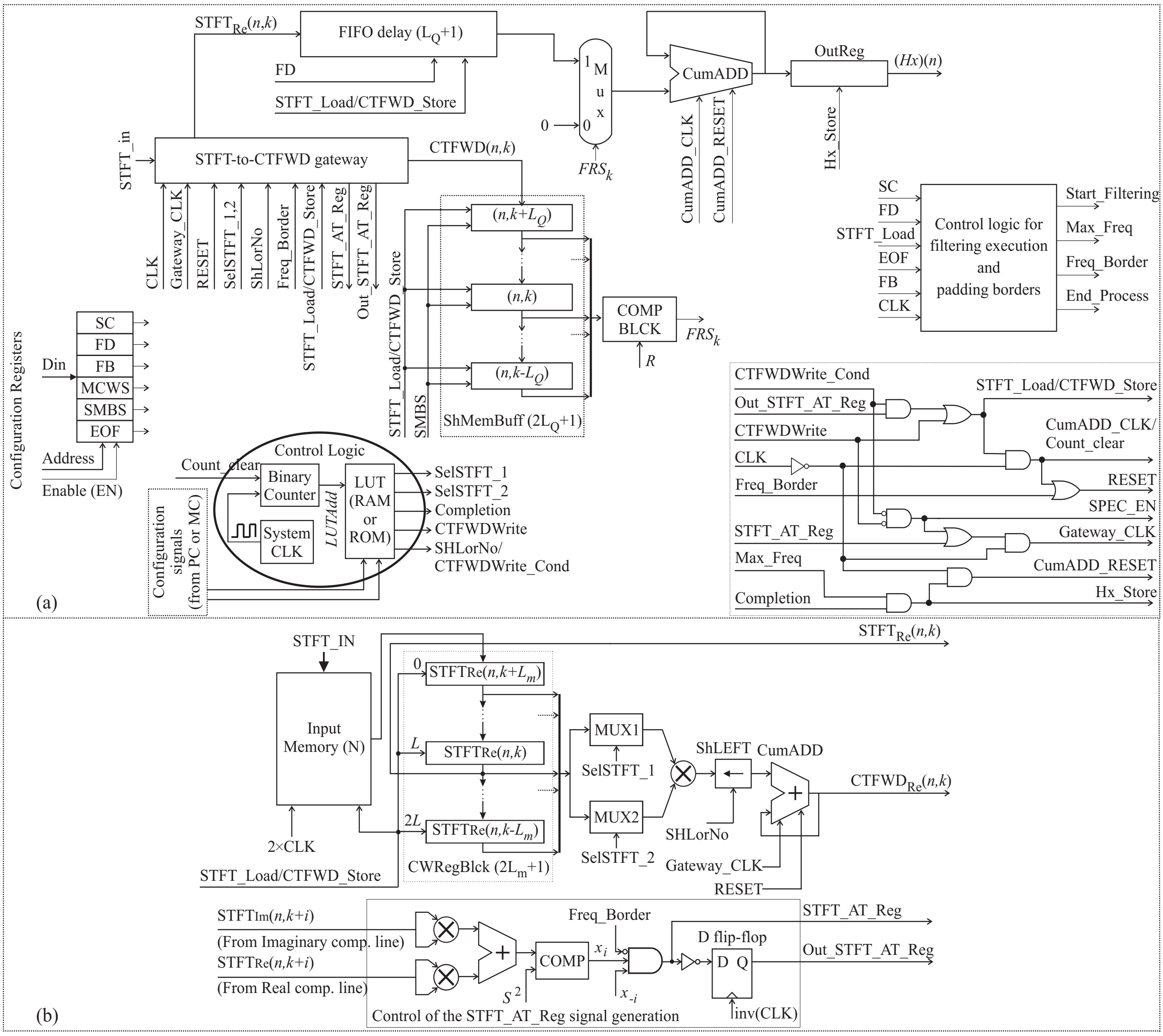

Figure 1 – (a) The proposed optimal nonstationary filtering system. In the centre of ShMemBuff registers, we denote the TF position of stored CTFWD samples in the sense of the estimation algorithm, proposed in [5]. (b) Real computational line of the STFT-to-CTFWD gateway.

[6]. L_m is the $L(n,k)$ maximum width, determined by the widest STFT auto-term. By definition, the CTFWD (3) is reduced to the SPEC ($|STFT_x(n,k)|^2$) outside STFT auto-terms' domains and to the WD inside them, taking desirable properties of these TFDs in the corresponding domains.

Further, CTFWD (3) consists of two identical computational lines (real and imaginary ones), used for processing of the STFT real and imaginary parts, respectively. Each of these lines takes form of eq.(3), where STFTs are replaced by their real or imaginary parts. Besides, in the case of real-valued signals, considered here, the STFT imaginary parts cancel each other in (1), so it becomes:

$$(Hx)(n) = \sum_{k=-N/2+1}^{N/2} L_H(n,k)\,\mathrm{Re}\{STFT_x(n,k)\}. \qquad (4)$$

Precisely, in the considered case of real-valued signals, $E\{CTFWD_x(n,-k)\}=E\{CTFWD_x{}^*(n,k)\}$ holds and, therefore, FRS $L_H(n,k)$ becomes a symmetrical function of frequency, [3, 5], that implies eq.(4).

3. SIGNAL ADAPTIVE HARDWARE IMPLEMENTATION

Architecture for the optimal TV filter real-time design is given in Fig.1. Following the real-time IF estimation algorithm proposed in [8] and using the already available CTFWD real-time design, [6], this architecture implements definition (1). Kernel of the proposed design, the STFT-to-CTFWD gateway, Fig.1.(b), implements algorithm (3). It modifies the input STFT data (STFT_IN) to produce an improved TF representation of the noisy signal based on the CTFWD, as discussed in [6].

The proposed hardware design performs the calculation in $L(n,k)$+3 CLKs per frequency point, Fig.2. In the first $L(n,k)$+1 CLKs, the CTFWD sample is calculated, [6]. The

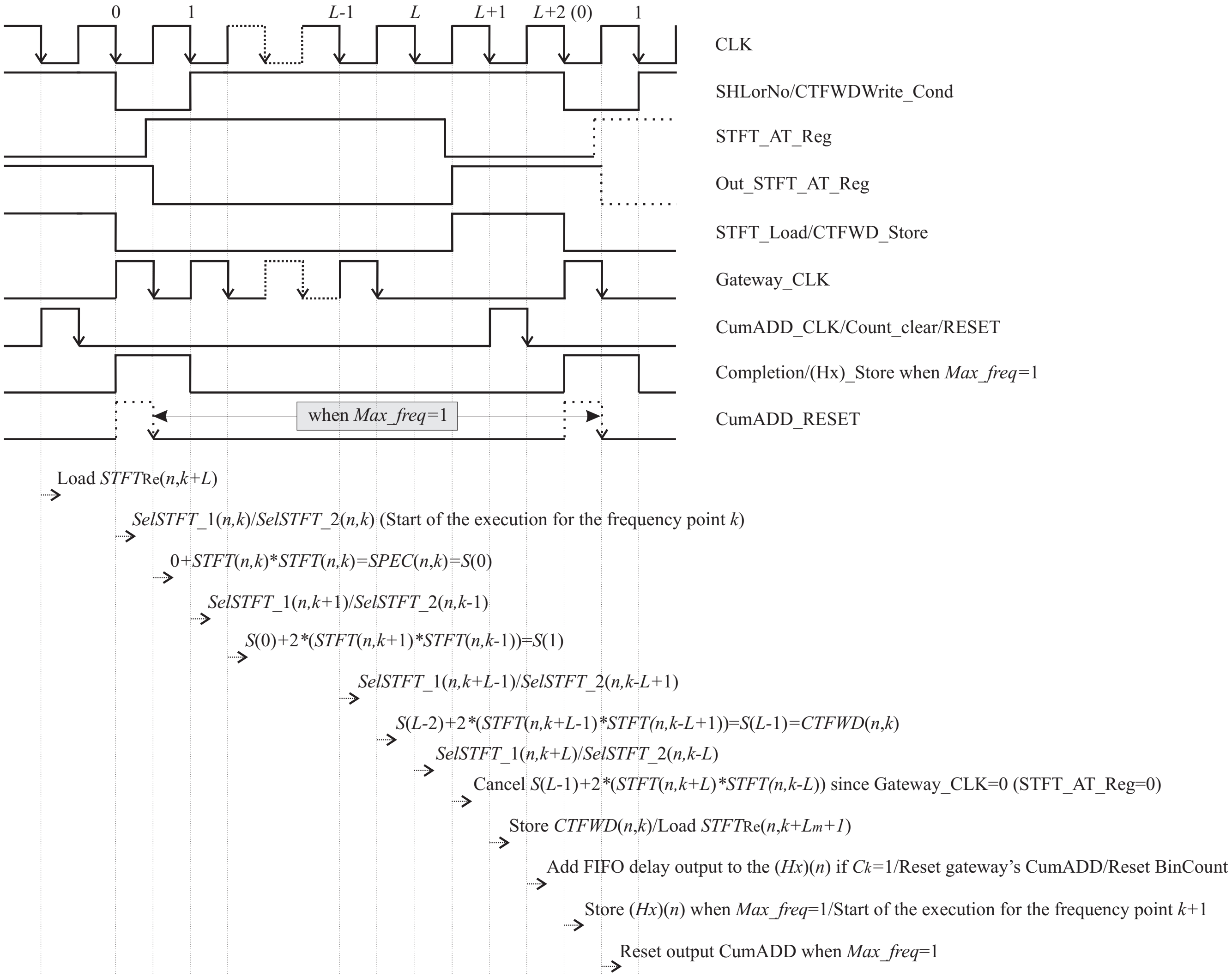
0
1
L-1
L
L+1
L+2 (0)
1
CLK
SHLorNo/CTFWDWrite_Cond
STFT_AT_Reg
Out_STFT_AT_Reg
STFT_Load/CTFWD_Store
Gateway_CLK
CumADD_CLK/Count_clear/RESET
Completion/(Hx)_Store when Max_freq=1
when Max_freq=1
CumADD_RESET
Load STFTRe(n,k+L)
SelSTFT_1(n,k)/SelSTFT_2(n,k) (Start of the execution for the frequency point k)
0+STFT(n,k)*STFT(n,k)=SPEC(n,k)=S(0)
SelSTFT_1(n,k+1)/SelSTFT_2(n,k-1)
S(0)+2*(STFT(n,k+1)*STFT(n,k-1))=S(1)
SelSTFT_1(n,k+L-1)/SelSTFT_2(n,k-L+1)
S(L-2)+2*(STFT(n,k+L-1)*STFT(n,k-L+1))=S(L-1)=CTFWD(n,k)
SelSTFT_1(n,k+L)/SelSTFT_2(n,k-L)
Cancel S(L-1)+2*(STFT(n,k+L)*STFT(n,k-L)) since Gateway_CLK=0 (STFT_AT_Reg=0)
Store CTFWD(n,k)/Load STFTRe(n,k+Lm+1)
Add FIFO delay output to the (Hx)(n) if Ck=1/Reset gateway's CumADD/Reset BinCount
Store (Hx)(n) when Max_freq=1/Start of the execution for the frequency point k+1
Reset output CumADD when Max_freq=1

Figure 2 – The execution – timing diagram in one time- and frequency-point for the hardware design from Fig.1.

TV filter function is then implemented in the next two CLKs $((L(n,k)+1)$-st and $(L(n,k)+2)$-nd ones), where the $(L(n,k)+2)$-nd one is overlapped in execution by the 0-th CLK of the next frequency point. The unconditional 0-th CLK, when $L(n,k)=0$, and the $(L(n,k)+1)$-st one provide the SPEC-based IF estimation in each TF point. Residual (conditional) CLKs (1-st, 2-nd, ..., $L(n,k)$-th one) are used to improve the IF estimation quality up to the CTFWD-based one, but only in TF points from the STFT auto-terms' domains, determined by the *STFT_AT_Reg* signal.

Signals $x_{\pm i}$, obtained at the comparator COMP output as: $x_{\pm I}=1$ if $|STFT(n,k\pm i)|^2 \geq S^2$ and $x_{\pm i}=0$ otherwise, determine non-zero values of $STFT(n,k\pm i)$ and generate the *STFT_AT_Reg* signal, $STFT\_AT\_Reg = x_i \times x_{-i}$ $(i=0,1,\ldots, L(n,k))$ in the corresponding CLKs. Through the participation in the *Gateway_CLK* signal generation, zero value of the *STFT_AT_Reg* signal disables the i-th term $(i=1,\ldots,L(n,k))$ to enter the summation (3) in i-th CLK, see Fig.2. The *Out_STFT_AT_Reg*=inv{*STFT_AT_Reg*} signal allows the *CTFWrite_Cond* signal, set in each conditional CLK, to make the unconditional $(L(n,k)+1)$-st cycle from the next conditional $((i+1)$-th) one. In this way, the *STFT_AT_Reg* signal allows the proposed design to optimize the number of CLKs taken in different TF points within the execution, to produce the CTFWD-based IF estimation (the highest quality one, [7]), and to significantly improve execution time in comparison to the non-adaptive MCI designs. Besides, the *STFT_AT_Reg* signal in combination with the *CTFWrite_Cond* signal control the filtering completion in the observed frequency point. The *SPEC_EN* signal provides execution of the unconditional 0-th cycle, even if $x_0=0$.

In the $(L(n,k)+1)$-st CLK, the computed CTFWD sample and real part of the corresponding STFT sample are stored respectively in the ShMemBuff (sized $2L_Q+1$ locations) and in the FIFO delay block (sized L_Q+1 locations), by setting the *STFT_Load/CTFWD_Store* signal. In parallel to this, the COMP BLCK generates the FRS_k signal, based on the actual content of ShMemBuff locations and on the real-time estimation algorithm from [5, 8]. It recognizes a local FRS, determined by $FRS_k=1$, in the frequency point that corresponds to the maximal ShMemBuff element, but only if that element is the central ShMemBuff element, and if it is greater than the introduced spectral floor R, [5, 8]. With the latency of half of a CLK, Fig.2, $FRS_k=1$ enables inclusion of the FIFO delay

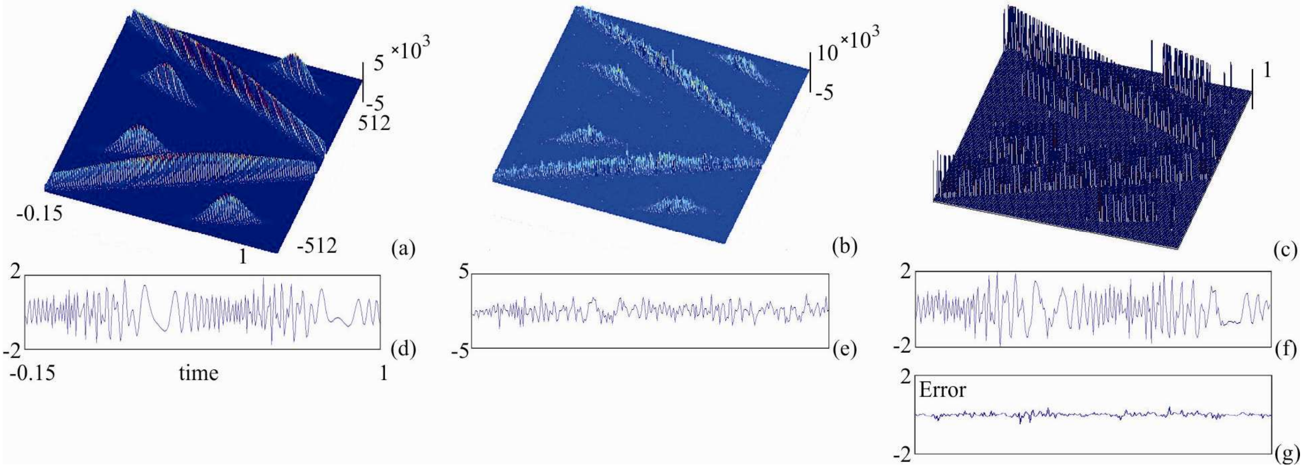

Figure 3 – (a) CTFWD of the non-noisy signal $f(t)$; (b) CTFWD of the noisy signal; (c) Estimated IF/FRS; (d) Signal $f(t)$; (e) Noisy signal; (f) Output signal of the proposed hardware design, implemented in real FPGA device EP1S10F780C5, (g) Filtering error.

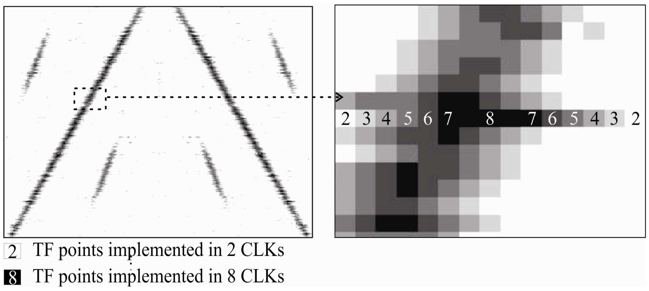

Figure 4 – Gray-scale shaded illustration (described principally by legend) of the number of CLKs taken by the signal adaptive design in corresponding TF points within the filtering of noisy signal (5).

output sample in the output signal $(Hx)(n)$ generation through the summation into the output cumulative adder (CumADD), whereas the RESET signal clears the STFT-to-CTFWD gateway. The $(L(n,k)+2)$-nd cycle is used for completion of the execution. However, the CumADD output will contain the final $(Hx)(n)$ value, for a given n, after performing the described execution in each frequency point in the observed n. It means that the final $(Hx)(n)$ value will be obtained after performing the execution in the maximum frequency point, i.e. when maximum frequency CTFWD sample becomes the central ShMemBuff element, detected by *Max_Freq* signal. Therefore, the *Completion* signal, generated in $(L(n,k)+2)$-nd cycle, has to be conditional one. It will store the calculated $(Hx)(n)$ value into the output register OutREG, but only when *Max_Freq*=1 is reached. With a latency of half of a CLK, the CumADD is reset and the execution for the next time-instant begins, see Fig.2. Simultaneously with $(L(n,k)+1)$-st CLK of the observed frequency point k, a new STFT_IN sample is imported and the described process is repeated for the next frequency point k+1. Note that importing of a new STFT_IN sample coincides with the *STFT_Load/ CTFWD_Store* cycle, whose period, therefore, must be $(L(n,k)+1)$ times greater than the CLK period. This implementation technique (pipelining) enables the overlapping in execution of the completion cycle of the considered frequency point k and the 0-th cycle of the next frequency point k+1. In this way we improve the total amount of work done in the given time rather than the execution time in an individual frequency point. This can be a significant development because ten thousand to several million described calculations (in different TF points) can be performed within the real filtering environment. The process is managed by the Look-up-table memory (LUT). Its locations consist of the 3-bit control signals area (*ShLorNo/ CTFWDWtite_Cond*, *CTFWDWrite*, *Completion* bits) and MUXs' addresses. The binary counter generates LUT's addresses. Operations at the maximal frequency are managed, as well as the values of L_Q and R parameters are set, in the same way as in the case of non-adaptive system [5].

4. TESTING AND VERIFICATION OF THE DESIGN

To verify the design, it has been implemented by using the EP1S10F780C5 device, from the Stratix II family. Before programming the selected device, the compilation and simulation have been performed by considering (within the time-interval [–0.15,1]) a sum of real-valued, highly nonstationary chirp signals (with very high normalized signal rates of 0.828, 0.844, 0.828, respectively), belonging to the commonly analyzed, wide class of finite duration FM signals:

$$f(t) = e^{-45(t-2/25)^2}\cos(900(t+1.3)^2) + e^{-(t-2/5)^2} \times\cos(1200(t+0.3)^2) + e^{-45(t-2/3)^2}\cos(900(t-1/22)^2) \quad (5)$$

where $t=nT_w/N$. Signal (5) is masked by the high white noise such that SNR_{in}=10×log(P_f/P_ε)=–0.34[dB]. The Hanning lag window width of T_w=0.25 has been applied, as well as $S^2=0.1\times\max_{n,k}\{SPEC_x(n,k)\}$, $R=0.05\times\max_{n,k}\{CTFWD_x(n,k)\}$, L_m=7, L_Q=5, and N=256. The efficiency of the proposed TV filter is evident, Fig.3. Very high improvement of SNR_{out}–SNR_{in}=17.37[dB] has been achieved in the considered case (theoretical SNR improvement of up to approximately (156/N)×10log(N/4)+(100/N)×10log(N/2)=19.2377 [dB] can be expected in the case of a partly 2-component signal (in 100 time-instants) and a partly 4-component signal (in 156 time-instants), Fig.3(a)-(c)).

Design	Hardware complexity		CLK cycle time	Execution time
	# of used funct. units	# of memory locations		
Parallel	$6L_m+5$	$4L_m+3L_Q+10$	$T_{cP}=2T_m+(L_m+4)T_a+$ $+T_s+T_{\mathrm{COMP\ BLCK}}$	$(N\times(N+1))\times T_{cP}$
Hybrid	$6L_m+5$	$4L_m+3L_Q+14$	$T_{cH}=2T_m+(L_m+3)T_a+T_s$	$(N\times N)\times 3T_{cH}$
Non-adaptive MCI	9	$5L_m+3L_Q+14$	$T_{cSF}=T_m+2T_a+T_s$	$(N\times N)\times(L_m+2)\times T_{cSF}$
Proposed	13	$2N+5L_m+3L_Q+13$	$T_{cSA}=2T_m+2T_a+2T_{comp}$	$144656\times T_{cSA}$

Table 1 – Hardware complexity, CLK cycle times and execution times of various implementations of the filtering definition (1). T_{cP}, T_{cH}, T_{cSF}, T_{cSA} are CLK cycle times in the cases of the parallel design, hybrid one, serial one with fixed number of CLKs, [5], and the signal adaptive one, respectively. $T_{COMP\ BLCK}$, T_m, T_a and T_s are the COMP BLCK, multiplication, addition and 1-bit shift times, respectively. Execution time of the proposed design has been given for the considered signal (5) case and N=256, L_m=7.

5. COMPARATIVE ANALYSIS AND CONCLUSIONS

In this Section, the proposed signal adaptive design will be explicitly compared with the other possible implementations of the considered estimation algorithm, Table 1. As well, it will be implicitly compared with the existing linear and nonlinear TV filtering solutions, [1–3], based on the comparisons (performed in [5]) of these solutions with the non-adaptive MCI solution [5]. Possible parallel and hybrid implementation approaches, considered in Table 1, would be based on the parallel implementation of the STFT-related gateway, proposed in [4]. Non-adaptive MCI approach with a fixed number of CLKs is based on the MCI of the STFT-based gateway, [5]. Then, the estimation of a local IF/FRS and the TV filter function would be implemented in the same CKL (case of the parallel approach), and in the next two CLKs (other cases), as described in Section 2. The proposed signal adaptive design almost achieves minimal hardware requirements and the CLK cycle time of the non-adaptive MCI design, optimizing these characteristics in comparison to the parallel and hybrid designs[2], and, therefore, in comparison to the existing linear and nonlinear filters, as discussed in detail in [5], when they are compared with the non-adaptive MCI design. In addition, the proposed design allows the implemented filter to take variable number of CLKs–the only necessary ones that provide CTFWD-based IF estimation quality–in different TF points within the execution, Fig.4: the minimal one outside the STFT auto-terms' domains (where the greater part of total TF points commonly lie), the higher one inside these regions, and the possible maximum one only around the central points of each STFT auto-term. In this way, the proposed design can significantly improve the execution time of other designs, removing the main drawback of the non-adaptive MCI architecture in comparison to the parallel and hybrid ones, as well as to the existing linear and nonlinear solutions, [1, 5]. For example, in the analyzed signal (5) case, when L_m=7, N=256 are applied, the proposed design execution time improves execution times of other corresponding designs (including the parallel design execution time) for $T_{\mathrm{COMP\ BLCK}}$, $T_{\mathrm{comp}}, T_s << T_m < 2.754\times T_a$. Finally, only the proposed design produces maximum quality (a pure CTFWD-based) IF/FRS estimation in the practically only important case of multicomponent signals having different STFT auto-terms widths. Non-adaptive approaches cannot produce so high estimation/filtering quality. For example, in the analyzed signal (5) case, we have numerically obtained the improvement of 15.56[dB] by using the considered non-adaptive approaches (versus the proposed design improvement of 17.37[dB], see Section 3).

[2] Parallel and hybrid designs minimize the total number of used memory locations, since the parallel one does not include LUT, whereas the hybrid one includes LUT of only three locations (controls the execution in 3 CLKs by frequency point). In addition, the proposed design includes two input memories (used for storing the real and imaginary parts of input STFTs), capacity of maximum N locations. However, note that the total number of used memory locations remains quite small in all considered cases.

REFERENCES

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

[2] G. F. Boudreaux-Bartels: "Time-varying signal processing using Wigner distribution synthesis techniques," in *The Wigner Distribution – Theory and Applications in Signal Processing* (W. Mecklenbräuker, F. Hlawatsch, eds.), Elsevier, 1997, pp.269–317.

[3] LJ. Stanković: "On the time-frequency analysis based filtering," *Ann. Telecomm.*, vol.55, May/June 2000, pp.216–225.

[4] S. Stanković, LJ. Stanković, V. N. Ivanović, R. Stojanović: "An architecture for the VLSI design of systems for time-frequency analysis and time-varying filtering," *Ann. Telecomm.*, vol.57, Sep/Oct.2002, pp.974–995.

[5] S. Jovanovski, V. N. Ivanović, "An efficient hardware design of an optimal nonstationary filtering system," in *Proc. IEEE Conf. ICASSP*, Taipei, Taiwan, April 19–24, 2009, pp.569–572.

[6] V. N. Ivanović, S. Jovanovski: "Signal adaptive system for time-frequency analysis," *Electron. Lett.*, vol.44, no.21, Oct.2008, pp.1279–1280.

[7] V. N. Ivanović, M. Daković, LJ. Stanković: "Performances of quadratic time-frequency distributions as instantaneous frequency estimators," *IEEE Trans. SP*, vol.51, no.1, Jan.2003, pp.77–89.

[8] S. Jovanovski, V. N. Ivanović, N. Radović: "An efficient real-time method for time-varying filter region of support estimation", *13th IEEE SPS DSP Workshop & 5th SPE Workshop*, Marco Island, USA, Jan. 4–7, 2009, pp. 513–517.