AN EFFICIENT SMART SYSTEM FOR IMPROVED SPACE/SPATIAL-FREQUENCY REPRESENTATION OF NONSTATIONARY 2-D SIGNALS

Veselin N. Ivanović, Srdjan Jovanovski

Dept. of Electrical Engineering, University of Montenegro
20000 Podgorica, MONTENEGRO
phone: +382 (20) 245 873, fax: +382 (20) 245 873, email: very@ac.me, srdjaj@ac.me

ABSTRACT

Multicycle implementation (MCI) of a signal adaptive two-dimensional (2-D) system for space/spatial-frequency (S/SF) analysis is proposed. The proposed design optimizes critical design performances related to the hardware complexity, making it a suitable system for real-time implementation on an integrated chip. It allows the implemented system to take a variable number of clock cycles (CLKs) (the only neces-sary ones regarding desirable–2-D Wigner distribution–presentation of auto-terms) in different frequency-frequency points during the execution. This ability represents the ma-jor advantage of the design which helps optimize the time required for execution and produce an improved, cross-terms-free S/SF signal representation. The implementation has been verified by an FPGA circuit design, capable of performing S/SF analysis of 2-D signals in real-time.

1.   INTRODUCTION AND BACKGROUND

Systems used in nonstationary 2-D signals processing are based on the developed mathematical methods, defined in their 2-D forms, [1–4]. These systems have been analyzed, usually in their parallel implementation forms, [5–7]. How-ever, such architectures (like the one based on the 2-D S-method, [5]) are quite complex, require duplication of basic calculation elements when they are employed more than once, and sometimes could not be implemented. In addition, the chip dimensions, power consumptions and cost are sig-nificantly increased in comparison to the corresponding 1-D architectures, [7–9], while the processing speed is decreased. At the same time, corresponding MCI designs of such sys-tems, [6], additionally decrease processing speed. Therefore, here we propose a way to overcome noted drawbacks of the 2-D parallel implementation and MCI forms.

The 2-D short-time Fourier transform (2-D STFT), its energetic version (2-D SPEC) and the 2-D Wigner distribu-tion (2-D WD) are conventional tools used in nonstationary 2-D signal analysis, [1–3], defined, in vector notation, as:

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

$$WD(\vec{n},\vec{k}) = \sum_{\vec{m}} w(\vec{m}) w(-\vec{m}) f(\vec{n}+\vec{m}) f^*(\vec{n}-\vec{m}) e^{-j2\frac{2\pi}{N}\vec{k}\vec{m}} \qquad (2)$$

where $w(\vec{m}) = w(m_1,m_2)$ is a 2-D real-valued window of

$N{\times}N$ duration, centered at the point $\vec{n} = (n_1,n_2)$ and used to truncate the analyzed signal $f(\vec{n})$. The 2-D STFT and 2-D SPEC provide cross-terms-free S/SF representation, but with a low concentration around signal's local frequency [2, 3]. On the other side, based on the direct 2-D STFT-to-2-D WD relationship, [2], readily following from (1)-(2),

$$
\begin{aligned}
WD(\vec{n},\vec{k}) = &\sum_{i_1=0}^{L_1} \Big(\mathrm{Re}\{STFT(\vec{n},k_1+i_1,k_2) STFT^*(\vec{n},k_1-i_1,k_2)\} \\
&+2\sum_{i_2=1}^{L_2^{(1)}} \mathrm{Re}\{STFT(\vec{n},k_1+i_1,k_2+i_2) STFT^*(\vec{n},k_1-i_1,k_2-i_2)\} \Big) \\
&+\sum_{i_1=1}^{L_1} \Big(\mathrm{Re}\{STFT(\vec{n},k_1+i_1,k_2) STFT^*(\vec{n},k_1-i_1,k_2)\} \\
&+2\sum_{i_2=1}^{L_2^{(2)}} \mathrm{Re}\{STFT(\vec{n},k_1+i_1,k_2-i_2) STFT^*(\vec{n},k_1-i_1,k_2+i_2)\} \Big)
\end{aligned}
\qquad (3)
$$

($L_1 = L_2^{(1)} = L_2^{(2)} = N/2$), the 2-D WD significantly improves the 2-D SPEC concentration (obtained for $i_1=i_2=0$), reaching the maximum concentration of each signal component sepa-rately and resulting in an optimal auto-terms' presentation, [2]. However, based on the full frequency-frequency domain 2-D convolution (3) of 2-D STFTs, the 2-D WD generates emphatic cross-terms in the multicomponent signals case.

To preserve 2-D WD auto-terms presentation and to si-multaneously reduce (or, completely eliminate, in the case of non-overlapping signal's components) 2-D WD cross-terms, the 2-D convolution in (3) should be performed inside the 2-D STFT auto-terms' domains (regions of support $D_i(\vec{n},\vec{k})$, i=1,...,q of a q-component signal), including only non-zero summation terms, and terminated outside these regions. Therefore, summation in (3) is limited by introduction of the rectangular 2-D convolution window with the signal adaptive width $\vec{L}(\vec{n},\vec{k}) = (L_1(\vec{n},\vec{k}), L_2(\vec{n},\vec{k},\pm i_1)) \le \vec{L}_m$, where $L_1 = L_1(\vec{n},\vec{k})$, $L_2^{(1)} = L_2(\vec{n},\vec{k},+i_1)$, $L_2^{(2)} = L_2(\vec{n},\vec{k},-i_1)$, and $\vec{L}_m$ is the maximum width of $\vec{L}(\vec{n},\vec{k})$, determined by the widest $D_i(\vec{n},\vec{k})$, i=1,...,q, and its corresponding local frequency $\vec{k}_{0i}$. For each point (k_1,k_2), this means, [4]: (i) The summation in $+i_2$ (limited by $L_2(\vec{n},\vec{k},+i_1)$), for each i_1, i_1=0, ..., $L_1(\vec{n},\vec{k})$,

2-D STFT — Algorithm (3) — **2-D CTFWD**

$2L_m+1$ L_m

2-D convolution window $2L_m+1$ $-N/2$

Path of conv. window L_m k_2 k_1 $N/2$-1 $-N/2$ $N/2$-1 (a)

k_2 k_1 $N/2$-1 $-N/2$ $N/2$-1

$(k_1\text{-}L_m,k_2\text{-}N/2)$	....	$(k_1\text{-}L_m,k_2\text{-}L_m)$	....	$(k_1\text{-}L_m,k_2)$	....	$(k_1\text{-}L_m,k_2+L_m)$	$(k_1\text{-}L_m,k_2+L_m+1)$	....	$(k_1\text{-}L_m,k_2+N/2\text{-}1)$
$(k_1,k_2\text{-}N/2)$	....	$(k_1,k_2\text{-}L_m)$	....	(k_1,k_2)	....	(k_1,k_2+L_m)	(k_1,k_2+L_m+1)	....	$(k_1,k_2+N/2\text{-}1)$
$(k_1+L_m,k_2\text{-}N/2)$	....	$(k_1+L_m,k_2\text{-}L_m)$	....	(k_1+L_m,k_2)	....	(k_1+L_m,k_2+L_m)	(k_1+L_m,k_2+L_m+1)	....	$(k_1+L_m,k_2+N/2\text{-}1)$

(b)

$STFT(\vec{n},k_1\text{-}L_m,k_2+L_m+1)$

$(2L_m+1)\times(2L_m+1)$ convolution window register block   **$2L_m$ FIFO delays**

| (k_1+L_m,k_2+L_m) | → | (k_1+L_m,k_2) | → | $(k_1+L_m,k_2\text{-}L_m)$ | $(k_1+L_m,k_2\text{-}L_m\text{-}1)$ | .... | $(k_1+L_m,k_2\text{-}N/2)$ | $(k_1+L_m\text{-}1,k_2+N/2\text{-}1)$ | .... | $(k_1+L_m\text{-}1,k_2+L_m+1)$ |

(k_1,k_2+L_m) → .... → (k_1,k_2) → .... → $(k_1,k_2\text{-}L_m)$ | $(k_1,k_2\text{-}L_m\text{-}1)$ | .... | $(k_1,k_2\text{-}N/2)$ | $(k_1+1,k_2+N/2\text{-}1)$ | .... | (k_1+1,k_2+L_m+1) |

$(k_1\text{-}L_m+1,k_2\text{-}L_m\text{-}1)$ .... $(k_1\text{-}L_m+1,k_2\text{-}N/2)$ $(k_1\text{-}L_m,k_2+N/2\text{-}1)$ .... $(k_1\text{-}L_m,k_2+L_m+1)$

$(k_1\text{-}L_m,k_2+L_m)$ → .... → $(k_1\text{-}L_m,k_2)$ → .... → $(k_1\text{-}L_m,k_2\text{-}L_m)$

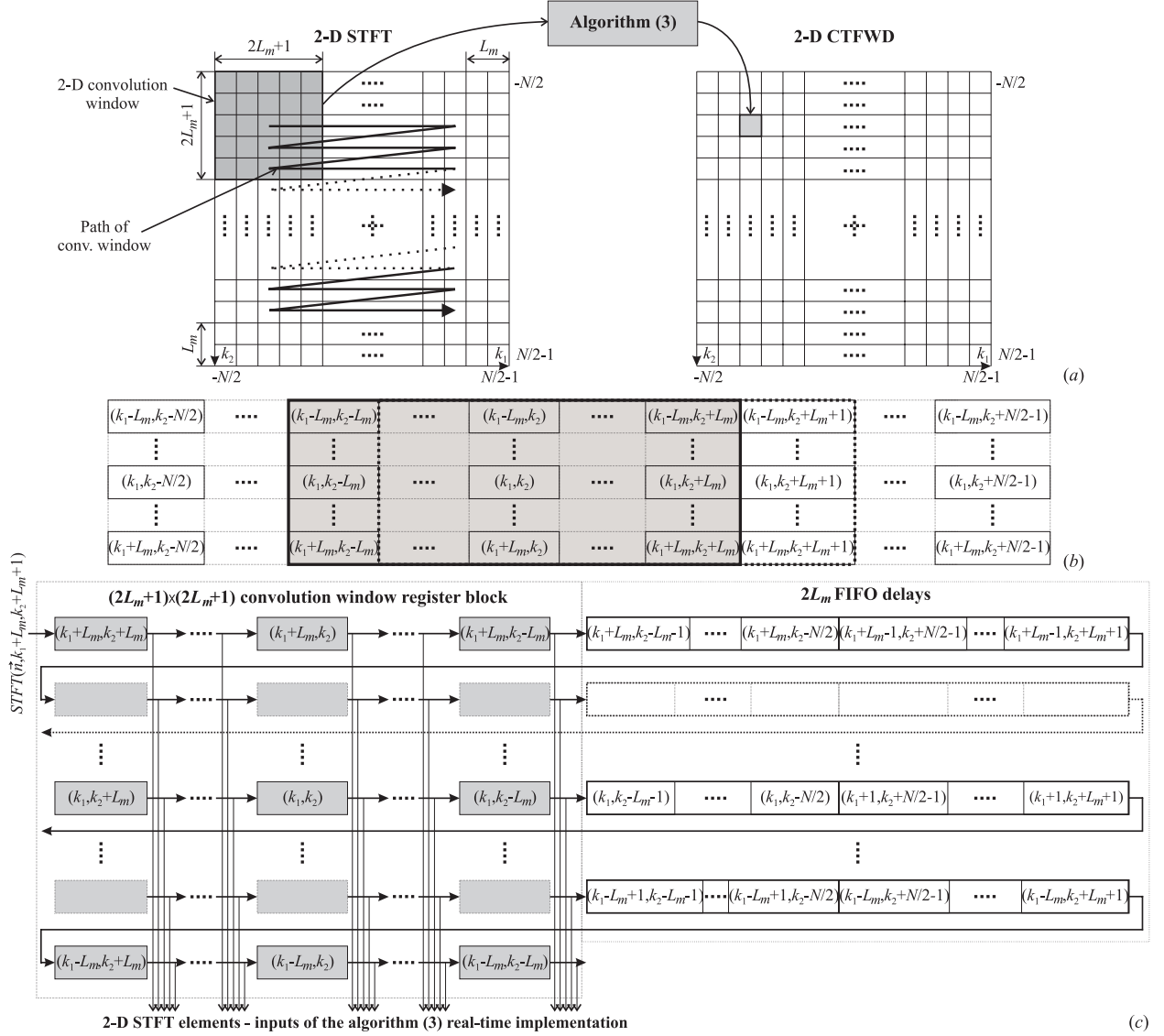**2-D STFT elements – inputs of the algorithm (3) real-time implementation** (c)

Figure 1: (a) Procedure of 2-D convolution window sliding and of 2-D CTFWD producing; (b) Actual convolution window position corresponding to the point (k_1,k_2) (thick solid line) and its next position (thick dashed line); (c); (d) Convolution window function, implemented in real-time. Cells correspond to the 2-D STFT elements, denoted by their position in frequency-frequency plane.

is performed until $|STFT(\vec{n},k_1\pm i_1,k_2\pm i_2)|^2 < R^2$, $i_2=0,\ldots,$ $L_2(\vec{n},\vec{k},+i_1)$, is detected[1], resuming the summation in $-i_2$; (ii) The summation in $-i_2$ (limited by $L_2(\vec{n},\vec{k},-i_1)$), for each i_1, $i_1=1,\ldots,$ $L_1(\vec{n},\vec{k})$, is performed until $|STFT(\vec{n},k_1\pm i_1,k_2\mp i_2)|^2 < R^2$, $i_2=0,\ldots,L_2(\vec{n},\vec{k},-i_1)$, is detected, resuming the summation for the next i_1; (iii) The summation in i_1 is performed until $|STFT(\vec{n},k_1\pm i_1,k_2)|^2 < R^2$, $i_1=0,\ldots,L_1(\vec{n},\vec{k})$, is detected, corresponding to the detection of the first zero

summation term not multiplied by 2 and resulting in completion of the calculation (3) in the point (k_1,k_2). This results in inclusion of the variable number of summation terms–the only necessary ones regarding the total energy of each autoterm separately–in different points (k_1,k_2), $k_1,k_2=0,1,\ldots,N-1$, reducing (3) to the 2-D SPEC outside $D_i(\vec{n},\vec{k})$, $i=1,\ldots,q$ (when $\vec{L}(\vec{n},\vec{k})=\vec{0}$) and to the 2-D WD inside them and producing the 2-D cross-terms-free WD (2-D CTFWD), [4].

2.   HARDWARE IMPLEMENTATION

In general, each 2-D CTFWD element is produced by sliding 2-D convolution window over the 2-D STFT input elements and by computing the 2-D CTFWD output value according to the input elements and the algorithm (3). The

---

[1] A predefined reference level R^2 (a few percent of 2-D SPEC's maximum value, [8]) determines regions D_i, $i=1,\ldots,q$, in the manner that 2-D STFTs whose absolute values are below R will be neglected in calculation.
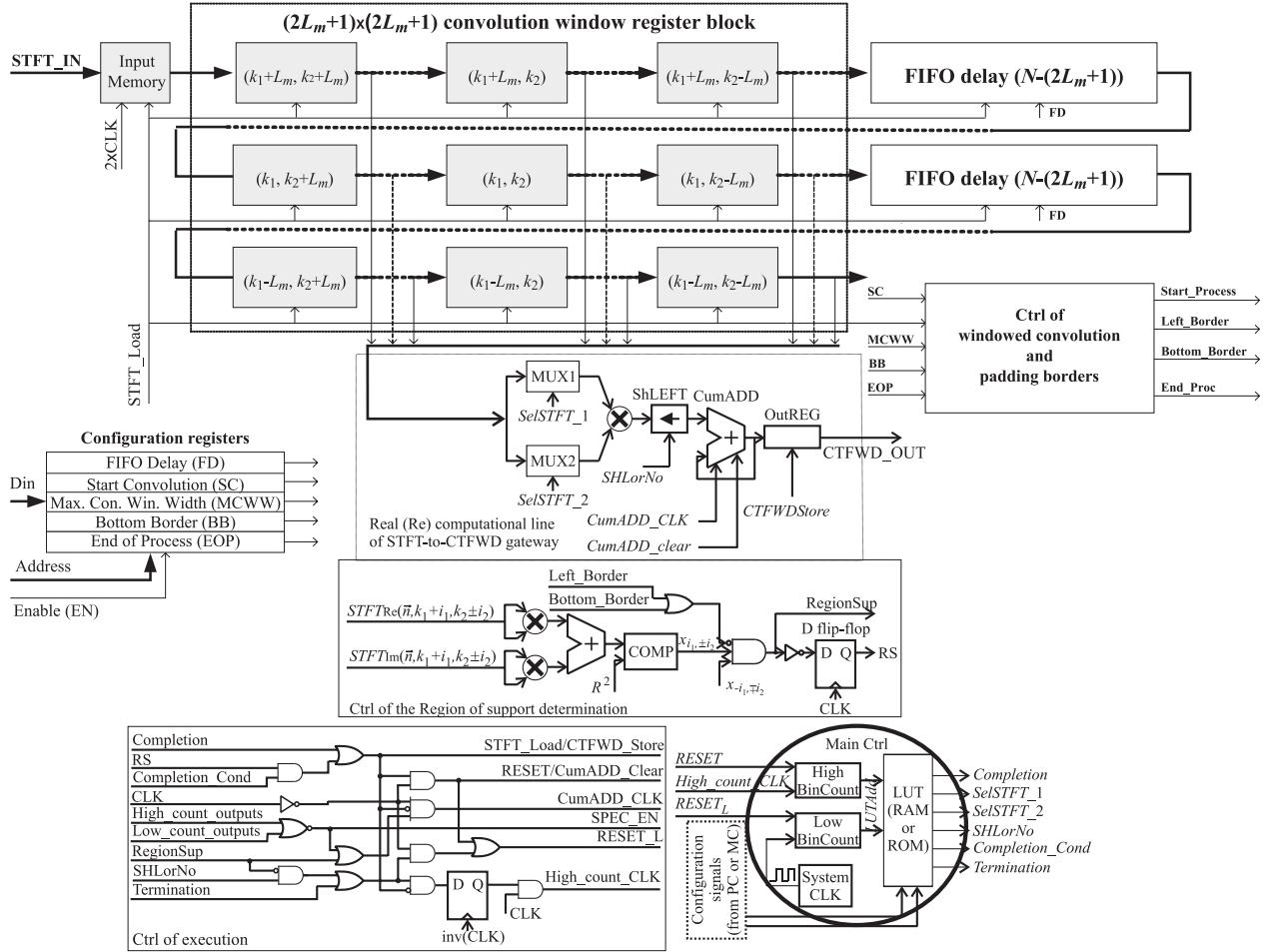
Figure 2: Proposed design for S/SF analysis. In the registers, the frequency-frequency positions of the stored 2-D STFT elements are denoted.

obtained result is a 2-D CTFWD element assigned to the center of the 2-D convolution window, Fig.1(a). Following these observations, the architecture for real-time design of the 2-D CTFWD has been implemented, Fig.2.

2-D CTFWD (3) adapted for real-time implementation should include only real multiplications. In that sense, we express it as a sum of two computational lines, used for processing the real and imaginary parts of 2-D STFT (from already available 2-D STFT or 2-D FFT modules, [3, 7, 8]). These lines have identical forms, obtained by replacing the 2-D STFTs with their real and imaginary parts in (3), respectively. Note that, in different points (k_1,k_2), $k_1,k_2=0,1,\ldots,N-1$, the 2-D CTFWD involves variable number of

$$CN(\vec{n},\vec{k}) = \sum_{i_1=1}^{L_1(\vec{n},\vec{k})} (L_2(\vec{n},\vec{k},+i_1) + L_2(\vec{n},\vec{k},-i_1) + 2) + L_2(\vec{n},\vec{k},0)$$

+1 summation terms. The hardware implementation will be presented through the implementation of the 2-D CTFWD real computational line and the control logic, Fig.2, since the imaginary computational line is identical with the real one, whereas the control units and configuration signals are unique. The design principle follows the form of eq.(3), where each summation term is executed during the corresponding step which takes one CLK. In this way, we are able

to balance the amount of work done in each CLK, resulting in minimization of the CLK cycle time, Table 1. During the first CLK, when $\vec{L}(\vec{n},\vec{k}) = \vec{0}$, the 2-D SPEC is executed from the $STFT(\vec{n},\vec{k})$ element. Residual summation terms, obtained for increased indices i_1 and/or i_2 in subsequent CLKs (second, third, …) and existing only in regions of support frequency-frequency points, are the conditional ones. They are used to improve the S/SFD concentration with the goal to achieve the one obtained by the 2-D WD. Therefore, the 2-D CTFWD real-time implementation requires variable number of CLKs by frequency-frequency point to be executed, where only 2-D SPEC execution CLK (the first one) remains the unconditional one in each point (k_1,k_2).

Real-time design, presented in Fig.2, consists of several main functional units. The STFT-to-CTFWD gateway represents a functional kernel of the proposed architecture. It is used to produce 2-D CTFWD outputs based on the 2-D STFT input elements obtained from the convolution window register block. The $(2L_m+1)\times(2L_m+1)$ convolution window register block and $2L_m$ first-in-first-out (FIFO) delays mutually implement the 2-D convolution window function. The convolution window register block determines the address
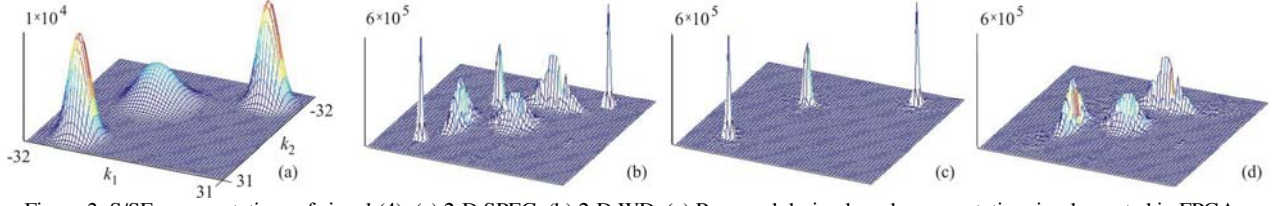
Figure 3: S/SF representations of signal (4): (a) 2-D SPEC, (b) 2-D WD, (c) Proposed design based representation, implemented in FPGA, (d) Difference between the proposed design signal (4) presentation (c) and the 2-D WD signal presentation (b).

order of the 2-D STFT input elements for which the corresponding 2-D CTFWD output will be computed according to (3). FIFO delays provide sliding of the 2-D convolution window over 2-D STFT input elements, presented in Fig.1(b).

The STFT_IN elements are imported to the input memory owing to each double CLK, corresponding to the minimal execution time by frequency-frequency point–2-D SPEC execution time. This period simultaneously determines sampling rate of the analyzed analogue 2-D signal. By each *STFT_Load* cycle, the STFT_IN elements are moved to the convolution window area that is sliced for one position right. According to the actual convolution window position and the algorithm (3), corresponding 2-D CTFWD output is calculated in $CN(\vec{n},\vec{k})+1$ CLKs. In each of the first $CN(\vec{n},\vec{k})$ CLKs, two of the convolution window register block elements, selected by the *SelSTFT*_1,2 signals, multiplied in the 2-input parallel multiplier and either doubled or not (depending on the *SHLorNo* signal), produce the corresponding summation term from (3) and the first input of the cumulative pipelined adder CumADD. Multiplying and shifting operations are parallel, while adding has a latency of half of a CLK. After $CN(\vec{n},\vec{k})$ CLKs, the CumADD output contains $CTFWD(\vec{n},\vec{k})$. The ($CN(\vec{n},\vec{k})+1$)-st CLK is the completion one. Therefore, the period of the *STFT_Load* cycle must be variable in different frequency-frequency points and $CN(\vec{n},\vec{k})+1$ times greater than the CLK period.

In the corresponding CLKs, signals $x_{\pm i_1, \pm i_2}$ and $x_{\pm i_1, \mp i_2}$ are generated as: $x_{\pm i_1, \pm i_2}=1$ if $|STFT(\vec{n},k_1 \pm i_1, k_2 \pm i_2)|^2 > R^2$ and $x_{\pm i_1, \pm i_2}=0$ otherwise, i.e. $x_{\pm i_1, \mp i_2}=1$ if $|STFT(\vec{n},k_1 \pm i_1, k_2 \mp i_2)|^2 > R^2$ and $x_{\pm i_1, \mp i_2}=0$ otherwise. They determine non-zero values of $STFT(\vec{n},k_1 \pm i_1, k_2 \pm i_2)$ and of $STFT(\vec{n}, k_1 \pm i_1, k_2 \mp i_2)$, $i_1 = 0,1,...,L_1(\vec{n},\vec{k})$, $i_2 = 0,1,...,L_2(\vec{n},\vec{k},i_1)$, respectively, and simultaneously produce the *RegionSup* control signal, $RegionSup = x_{i_1, \pm i_2} \cdot x_{-i_1, \mp i_2}$. Zero value of the *RegionSup* signal implies following actions: (1) Through the participation in the *CumADD_CLK* signal generation, disables corresponding term, non-existing into the regions of support domains, to enter the summation (3), and (2) Through the participation in the *RESET_L* and the *High_Count_CLK* signals generation, terminates the summation in $+i_2$ and in $-i_2$ (for each i_1, $i_1 = 0,1,...,L_1(\vec{n},\vec{k})$), resuming the summation in $-i_2$ and for the next i_1, respectively. Further, zero value of the *RegionSup* signal, reached in the

CLK cycle when *SHLorNo*=0 (corresponding to the summation term from (3), not multiplied by 2), terminates the summation (3) in i_1, resulting in the calculation completion (in the next CLK) for the observed point (k_1,k_2). In that sense, the *RS* signal, *RS*=inv(*RegionSup*), participates in the *STFT_Load/CTFWD_Store* signal generation. With a latency of half of a CLK, the CumADD, High_Bin_Count and Low_Bin_Count are reset and the execution for the next frequency-frequency point begins. In this way, the *RegionSup* signal allows the proposed design to optimize the number of CLKs taken in different frequency-frequency points within the execution. The *SPEC_EN* signal provides execution of the first (2-D SPEC execution) cycle, even if $x_{0,0}=0$.

A look-up-table (LUT) manages the execution. Its locations consist of the 4-bit control signals area (*SHLorNo*, *Completion_Cond*, *Termination*, *Completion* bits) and MUX addresses. Through the participation in the *STFT_Load/ CTFWD_Store* and *CumADD_Clear/RESET* signals generation, the *Completion_Cond* signal allows the *RS* signal to produce the completion cycle from the conditional one, when *RS*=1 is reached. The *Termination* and *Completion* signals provide termination of the summation (3) in $+i_2$ and in $-i_2$ and its completion, respectively, in points (k_1,k_2) that theoretically require maximum convolution window width greater than a predefined $(2L_m+1) \times (2L_m+1)$ (in practical implementations, L_m has to be predefined and, therefore, can be smaller than the theoretically required one in points (k_1,k_2) existing around the local frequency). In these points, the *RegionSup* signal cannot achieve zero value and, therefore, the *Termination* and *Completion* signals are introduced to assume its role. Further, the desired–2-D WD–concentration is already reached with relatively small L_m, $5 \le L_m \le 7$, which, on the other hand, significantly simplify hardware implementation, Fig.2. Binary counter Low_Bin_Count generates LUT's low addresses, controlling the summation (3) in $+i_2$ and in $-i_2$. Binary counter High_Bin_Count sets LUT's high addresses, controlling the summation (3) in i_1. Operations at the bordering positions, as well as the whole S/SF process, are managed by *Start_Process*, *Left_Border*, *Bottom_Border* and *End_Process* signals. Through the participation in the *RegionSup* signal generation, the *Left_Border* and *Bottom_Border* signals allow padding the left and bottom borders with $2L_m$ 2-D SPECs.

The longest path that determines the fastest CLK time corresponds to the generation of the *RegionSup* signal in a half of a CLK, through a multiplier, an adder and a comparator ($T_c/2=T_m+T_a+T_{comp}$, where T_c, T_m, T_a, T_{comp} are CLK, multiplication, addition and comparison times, respectively). In this way, the participation of the *RegionSup* signal in the

Table 1. CLK cycle times and execution times (by frequency-frequency point) of the considered implementations of 2-D systems. T_{cP}, T_{cSF}, T_{cSA} are CLK cycle times in the cases of the parallel design, MCI one with a fixed number of CLKs and the signal adaptive one, respectively, whereas T_s is the 1-bit shift time. Proposed design execution time has been calculated for the considered signal (4) and $N=64$, $L_m=5$.

Implementation	Clock cycle time	Execution time
Parallel (when it is possible)	$T_{cP} = 2T_m + (2L_m^2 + 2L_m + 2)T_a + T_s$	T_{cP}
Serial with a fixed number of CLKs	$T_{cSF} = T_m + 2T_a + T_s$	$(2L_m^2 + 2L_m + 2)T_{cSF}$
Proposed signal adaptive	$T_{cSF} = 2T_m + 2T_a + 2T_{comp}$	$8.6245 \times T_{cSA}$

CumADD_CLK signal generation in the second half of the same cycle is enabled.

3. TESTING AND VERIFICATION

The proposed design will be verified by considering the 2-D signal

$$f_1(n_1,n_2) = \cos[20\pi(n_1T - 0.75)^2 + 22\pi(n_2T - 0.75)^2]$$

$$+ 0.5\exp(j[-100\cos(\pi n_1 T/2) + 100\cos(\pi n_2 T/2)]) \quad (4)$$

$$f_2(n_1,n_2) = \cos\{1000\pi[(n_1T + 0.5)^2 + (n_2T - 0.5)^2]\}.$$

consisting of dual components: infinite duration $f_1(n_1,n_2)$, considered in the range $|n_1T|<0.75$, $|n_2T|<0.75$, and $f_2(n_1,n_2)$ having comparatively small domain of $|n_1T+n_2T|<0.1$, $|n_2T-n_1T-1|<0.1$. A sampling interval of $T=1/64$, the Hanning lag window $w(\vec{m})$, $N=64$, $L_m=5$ and R^2 of the 1% of the SPEC's maximum value have been used. For the proposed design realization, the EP2S15F672C5 device, from the Stratix II family, has been used. Results, computed at the point $(n_1T,n_2T)=(-0.25,-0.25)$, are presented in Fig.3. Low 2-D SPEC's concentration, Fig.3(a), high 2-D WD resolution, but with the emphatic cross-terms presence, Fig.3(b), as well as the optimal auto-terms presentation, but also the pure cross-terms-free S/SF signal (4) representation, Fig.3(c), obtained by the proposed design, can be readily noticed. The pure 2-D CTFWD signal representation, reached by the proposed design and given in Fig.3(c), is readily proven by Fig.3(d).

4. COMPARATIVE ANALYSIS AND CONCLUSIONS

To underline significance of the proposed MCI signal adaptive design, the comparison with the other implementations of systems for S/SF signal analysis (the possible parallel one with a fixed CLK cycle, [5], and the existing MCI one with a fixed number of CLKs, [6]) will be performed. Trade-offs and comparisons of the considered implementation approaches with respect the time required for execution are summarized in Table 1. In general, MCI designs imply both minimal hardware requirements and much shorter CLK cycle time in comparison to the parallel design. On the other side, MCI designs require greater time for execution. However, the proposed signal adaptive design allows the implemented S/SF system to take variable number of CLKs–the only necessary ones that provide CTFWD signal representation quality–in different frequency-frequency points within the execution: the minimal one outside the regions of support (where the greater part of total frequency-frequency points commonly lie, [1, 2, 6]), the higher one inside these regions, and the possible maximum one only around the central points of each region of support. In this way, the proposed design can significantly improve the execution time in comparisons to the other designs, removing the main drawback of the MCI architectures in comparison to the parallel ones, [7–9]. For example, in the analyzed signal (4) case, when $L_m=5$, $N=64$ are applied, the proposed design execution time improves execution times of other corresponding designs under very comfortable hardware demand $T_s,T_{comp}<<T_m<2.935\times T_a$. Finally, only the proposed design produces a pure 2-D CTFWD signal representation in the practically only important case of the multicomponent signals having different 2-D STFT auto-terms widths. Systems, based on the non-adaptive S/SF methods, cannot produce so high S/SF representation quality (see Fig.3, as well as [1, 4–6]). These abilities qualify the proposed design to become an optimal solution for real-time processing of wide range of nonstationary 2-D signals.

REFERENCES

[1] L. Jacobsen, H. Wechsler: "Joint spatial/spatial-frequency representation," *Signal Processing,* vol.14, 1988, pp.37–68.

[2] S. Stanković, LJ. Stanković, Z. Uskoković: "On the local frequency, group shift and cross-terms in the multidimensional time-frequency distributions," *IEEE Trans. SP*, vol.43, no.7, 1995, pp.1719–2661.

[3] A. Papoulis, *Signal Analysis*, New York: McGraw-Hill Book Co., 1977.

[4] V.N. Ivanović, S. Jovanovski: "Signal adaptive method for improved space/spatial-frequency representation," *Electronics Letters*, vol.45, no.19, Sept.2009, pp.1003–1005

[5] S. Stanković, I. Djurović, V. Vuković, "System architecture for space-frequency image analysis," *Electronics Letters*, vol.34, no.23, Nov.1998, pp.2224-2225.

[6] V.N. Ivanović, R. Stojanović, "An efficient hardware design of the flexible 2-D system for space/spatial-frequency signal analysis," *IEEE Trans. SP,* vol. 55, no. 6, 2007, pp. 3116–3125.

[7] K.J.R. Liu, "Novel parallel architectures for short–time Fourier transform," *IEEE Trans CAS-II*, vol.40, no.12, 1993, pp.786–789.

[8] S. Stanković, LJ. Stanković, "An architecture for the realization of a system for time-frequency analysis," *IEEE Trans. CAS-II*, vol. 44, no. 7, 1997, pp. 600–604.

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