MULTIPLE CLOCK CYCLE REAL-TIME IMPLEMENTATION OF A SYSTEM FOR TIME-FREQUENCY ANALYSIS

Veselin N. Ivanović, LJubiša Stanković

Dept. of Electrical Engineering, University of Montenegro,
81000 Podgorica, MONTENEGRO
phone: +381 81 244 921, fax: +381 81 244 921, email: very@cg.ac.yu, l.stankovic@ieee.org
web: http://www.tfsa.cg.yu

ABSTRACT

Multiple clock cycle implementation of a flexible system for time-frequency (TF) signal analysis is presented. It allows TF distributions (TFDs) to take different numbers of clock cycles and to share functional units within a TFD execution. These abilities represent the major advantages of multicycle design and they help reduce both hardware complexity and cost. The designed hardware is suitable for a wide range of applications, because it allows sharing in realization of some frequently used TFDs: Spectrogram (SPEC), S-method (SM) for various convolution window widths, and pseudo Wigner distribution (WD), as well as for the realization of the higher order TFDs.

1. INTRODUCTION

In order to alleviate the serious drawbacks of commonly used TFDs: the SPEC and WD, the SM is defined in [7], and intensively used in [3, 6, 8]. It is defined based on the short-time Fourier transform (STFT). This fact makes it very attractive for implementation. However, all TFDs, beyond the STFT, are numerically quite complex and require significant calculation time. This fact makes them unsuitable for real-time analysis, and severely restricts their application. Hardware implementations, when they are possible, can overcome this problem and enable application of these methods in numerous additional problems in practice. A simple implementation of the architectures for VLSI design of the systems for TF analysis and time-varying filtering based on the SM is represented in [10, 11]. They give desired TFD in one clock cycle. This means that no architecture resource can be used more than once, and that any element needed more than once must be duplicated. Besides, the single TFD - SM with exactly defined convolution window width - can be realized by these architectures.

In this paper we develop a multiple clock cycle implementation of a flexible system for TF analysis, based on the SM, and suitable for VLSI design. In this implementation, each step in the execution will take one clock cycle. In the first step, proposed architecture realizes the STFT. In each higher order step, different TFD is realized: in the second one - the SPEC, in the third one - the SM with unitary convolution window width, and so on. In further steps this architecture can realize the second order L-Wigner TFD (LWD) based on the SM realization in the preceding steps.

This implementation allows a functional unit to be used more than once per TFDs execution, as long as it is used on different clock cycles. This will help to significantly reduce the amount of the required hardware, and to reduce its cost.

2. REVIEW OF THE IMPLEMENTED TFDs

S-method is defined based on the unified definition of the STFT and the WD in the following manner [7, 9]:

$$SM(n,k)=\sum_{i=-L_d}^{L_d} P(i)F(n,k+i)F^*(n,k-i) \qquad (1)$$

where $F(n,k)$ represents the STFT of the analyzed signal $f(n)$, whereas $W_P=2L_d+1$ is a width of the convolution window $P(i)$ and the signal's duration is $N=2^m$. For the marginal cases of the $P(i)$ width: *a*) $L_d=N/2$, the WD is obtained, *b*) $L_d=0$, the SPEC follows. By an appropriate selection of the $P(i)$ width, the SM produces better results than the SPEC and the WD, regarding some the most essential aspects, [7, 9]. In order to involve only real multiplications in (1), we modify it by using $F(n,k)=F_{\mathrm{Re}}(n,k)+jF_{\mathrm{Im}}(n,k)$ ($F_{\mathrm{Re}}(n,k)$ and $F_{\mathrm{Im}}(n,k)$ are the real and imaginary part of $F(n,k)$, respectively), as:

$$SM_R(n,k)=F_{\mathrm{Re}}^2(n,k)+2\sum_{i=1}^{L_d} F_{\mathrm{Re}}(n,k+i)F_{\mathrm{Re}}(n,k-i) \qquad (2)$$

$$SM_I(n,k)=F_{\mathrm{Im}}^2(n,k)+2\sum_{i=1}^{L_d} F_{\mathrm{Im}}(n,k+i)F_{\mathrm{Im}}(n,k-i) \qquad (3)$$

where $SM(n,k)=SM_R(n,k)+SM_I(n,k)$. The k-th channel, one of N channels (obtained for k=0,1,...,N-1), is described by (2)-(3). Note that it will consist of two identical sub-channels used for processing of $F_{\mathrm{Re}}(n,k)$, and $F_{\mathrm{Im}}(n,k)$, respectively.

Higher order TFDs are used to improve concentration of the highly nonlinear FM signals. One of them, the LWD of the L-th order, can be defined in a recursive manner as, [8]:

$$LW_L(n,k)=\sum_{i=-L_d}^{L_d} LW_{L/2}(n,k+i)LW_{L/2}(n,k-i) \qquad (4)$$

where $LW_1(n,k)=SM(n,k)$. Since (4) is of the same form as (1), the LWD can be realized by using the same hardware.

Total number of	**SCI**	**MCI**
Adders	$2L_d + 1$	3
Multipliers	$2(L_d + 1)$	2
Shift Left Reg.	$4L_d$	2

Table 1: Total number of functional units per channel in an SM block, in the cases of single-cycle implementation (SCI) and the multicycle implementation (MCI).

3. MULTICYCLE IMPLEMENTATION

The hardware necessary for one channel multicycle implementation of the SM is presented in Fig.1. It is designed based on the two-block structure. The first block is used for the STFT implementation, whereas the second block is used to modify the outputs of the STFT block in order to obtain the improved TFD concentration based on the SM (or based on LWD). The STFT block is implemented based on the recursive algorithm, [1, 4, 5, 11], since, due to the reduced hardware complexity, it is more suitable for VLSI implementation. The SM block is designed so that it realizes each summation term from the eqs.(2)-(4) in the corresponding step of the TFD implementation.

Our goal in breaking the TFDs execution into clock cycles should be to balance the amount of work done in each cycle, so that we minimize the clock cycle time. We will break the execution into several steps, each taking one clock cycle. In the first step, the STFT will be executed, in the second step the SPEC will be executed based on the first step execution, in the third step the SM with the unitary convolution window width will be executed based on the execution in first two steps, and so on. With each larger step one realizes the SM with the incremental value of convolution window, based on preceding steps. This improves the TFD concentration, aiming to achieve the one obtained by the WD. Note that in the first step only the STFT block of the proposed two-block architecture is used, whereas in the other steps only the SM block is used. This will be regulated by the set of control signals introduced on temporary registers, and multiplexers and a demultiplexer.

Each sub-channel of the SM block contains exactly one adder, one multiplier, and one shift left register for implementation of eqs.(2)-(3). Because we need to use these functional units with different inputs in later steps, we must save the computed values, based on eqs.(2)-(3), into a temporary registers named *Real* and *Imag*, respectively. In addition, the SM block is used for other purposes in steps in which the higher order TFD is realized. Than we must save the computed SM into a new temporary register named *SMStore*. In order to share functional units from the SM block, as well as the very SM block, for different inputs in different clock cycles, we need to add multiplexers and/or a demultiplexer at their inputs. The introduced (de)multiplexers are:

- A two-input multiplexer at the input of the SM block with the control line *SMorLWD*. It enables sharing of the SM block for the LWD realization;
- Two $N/2$-input multiplexers at multiplier's inputs to select between the $N/2=2^{m-1}$ sources of STFT values - from different channels of proposed architecture. They are controlled by (m-1)-bit control signal *SelSTFT*;
- One two-output demultiplexer in front of the Shift Left register and one two-input multiplexer behind it, enable use of the Shift Left register in corresponding steps (third, fourth, and so on, when we need to implement multiplication by 2). Control signal *SHLorNo* enables or disables (in the second - SPEC completion step) this;
- A two-input multiplexer for sharing the second adder input in each sub-channel. Since one adder per sub-channel is used for implementing sums in eqs.(2)-(3), its second input can be either the constant 0 (in second step) or a register *Real* (or *Imag*) value, depending on the control signal *AddSelB*;
- A two-input multiplexer at the second input of the output adder in the SM block, which is controlled by the *SMorLWD* signal. Namely, when the SM block realizes the higher order TFD, it processes $SM(n,k)$, and only half of this block is used, since the SM is always real, [7]-[9].

Besides the multiplexers and demultiplexer controls, we will need to add four 1-bit signals for controlling writes in the registers R1 to R64, and in temporary registers:

- *SignLoad* enables sampling of the analyzed analog signal $f(t)$ and loading these samples in registers R1 to R64, but only after execution of the desired TFD of the analyzed signal samples from the preceding time instant;
- *RealWrite* and *ImagWrite* should be asserted only when a registers *Real* and *Imag* are to be written by the adder's output;
- *SMWrite* should be asserted when the SM with corresponding convolution window width is computed.

By introducing the temporary registers and several multiplexers at the inputs of the functional units, we achieve the required reduction of the amount of hardware compared to a single-cycle architecture, [10, 11] and Table I. The achieved hardware reduction is significant, and it increases with the convolution window width increase. Since temporary registers and introduced multiplexers are fairly small, this could yield a substantial reduction in the hardware cost.

The throughput of the system is N. The longest path in the STFT block is one that connects the register storing $F_{\mathrm{Re}}(n\text{-}1,k)$ (or $F_{\mathrm{Im}}(n\text{-}1,k)$), through one multiplier and 2 adders, with the output of the STFT block. It is the longest path in the SM block, as well. This path determines the fastest sampling rate and the clock cycle time. This design can be implemented as an ASIC chip to meet the speed and performance demands of very fast real-time applications.

Defining the Control. From the defined multi-step sequence of the multicycle TFDs execution we can determine what the Control logic must do at each clock cycle. It can set all control signals, based solely on the distribution code (TFDcode). Control for the multicycle architecture must specify both the signals to be set in any step and the next step in the sequence. Here, we use finite state Moore machine to specify the multicycle control, Fig.2. The implementation of a finite state machine usually assumes that all outputs, that

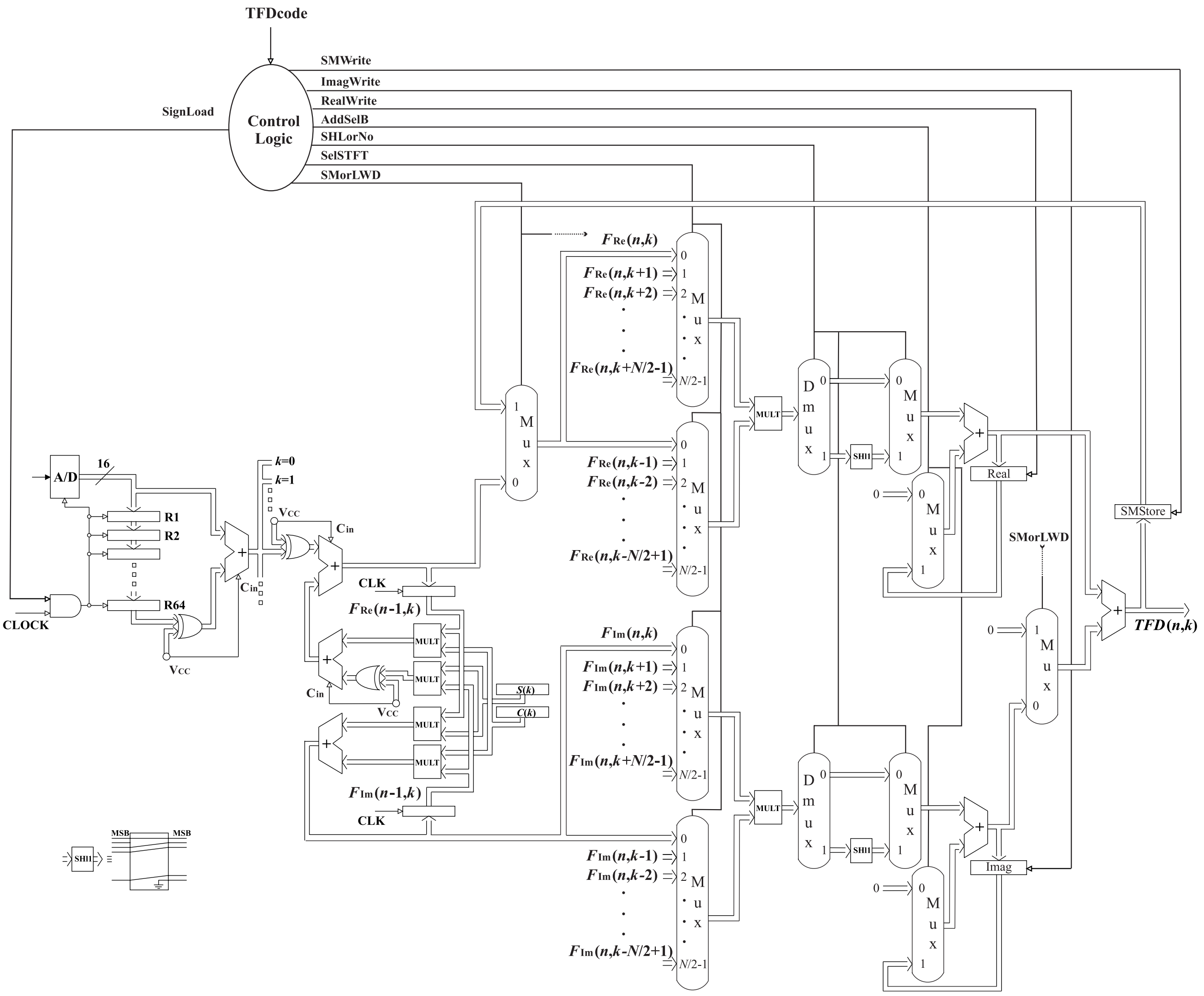

Figure 1: Architecture for the multicycle implementation of the signal independent S-method together with the necessary control lines. Thick solid line highlights the control line as opposed to a line that carries data.

are not explicitly asserted, are deasserted. Finite state Control essentially corresponds to the steps of desired TFD execution; each state in the finite state machine will take one clock cycle. In the first L_d+2 clock cycles, system realizes *SM*(n,k). At the same time, the control signal *SMorLWD* takes zero value. The calculated SM value is saved in the *SMStore* register, by asserting *SMWrite* control signal. Saved SM value will be used in the next L_d+1 clock cycles, when the LWD with L=2 will be realized. In these clock cycles the control signal *SMorLWD* is asserted, in order to enable the *SM*(n,k) processing in the SM block. Note that if we repeat the last L_d+1 steps from Fig.2 (i.e., steps L_d+2 to 2L_d+2), together with asserting of the *SMWrite* control signal in the (2L_d+2)-th step, the LWD with L=4 is implemented, as well.

4. CONCLUSION

The system for multiple clock cycle implementation of the TF algorithms is presented. Proposed hardware is very flexible since it can produce some commonly used methods, each of them in different clock cycle. In higher clock cycles it may produce the higher order TFDs, as well. The proposed implementation allows a functional unit to be used more than once per TFDs execution, as long as it is used on different clock cycles, and, consequently, enables a significant reduction of hardware complexity.

REFERENCES

[1] M.G. Amin: “A new approach to recursive Fourier transform,” *Proc. IEEE*, vol.75, Nov.1987, pp.1357-1358.

[2] L. Cohen, *Time-frequency analysis*, Prentice Hall, 1995.

[3] P. Goncalves, R.G. Baraniuk: “Pseudo affine Wigner distributions: Definition and kernel formulation,” *IEEE Trans. SP*, vol.46, no.6, June 1998, pp.1505-1517.

[4] K.J.R. Liu: “Novel parallel architectures for Short-time Fourier transform,” *IEEE Trans. CAS-II*, vol.40, no.12, Dec.1993, pp.786-789.

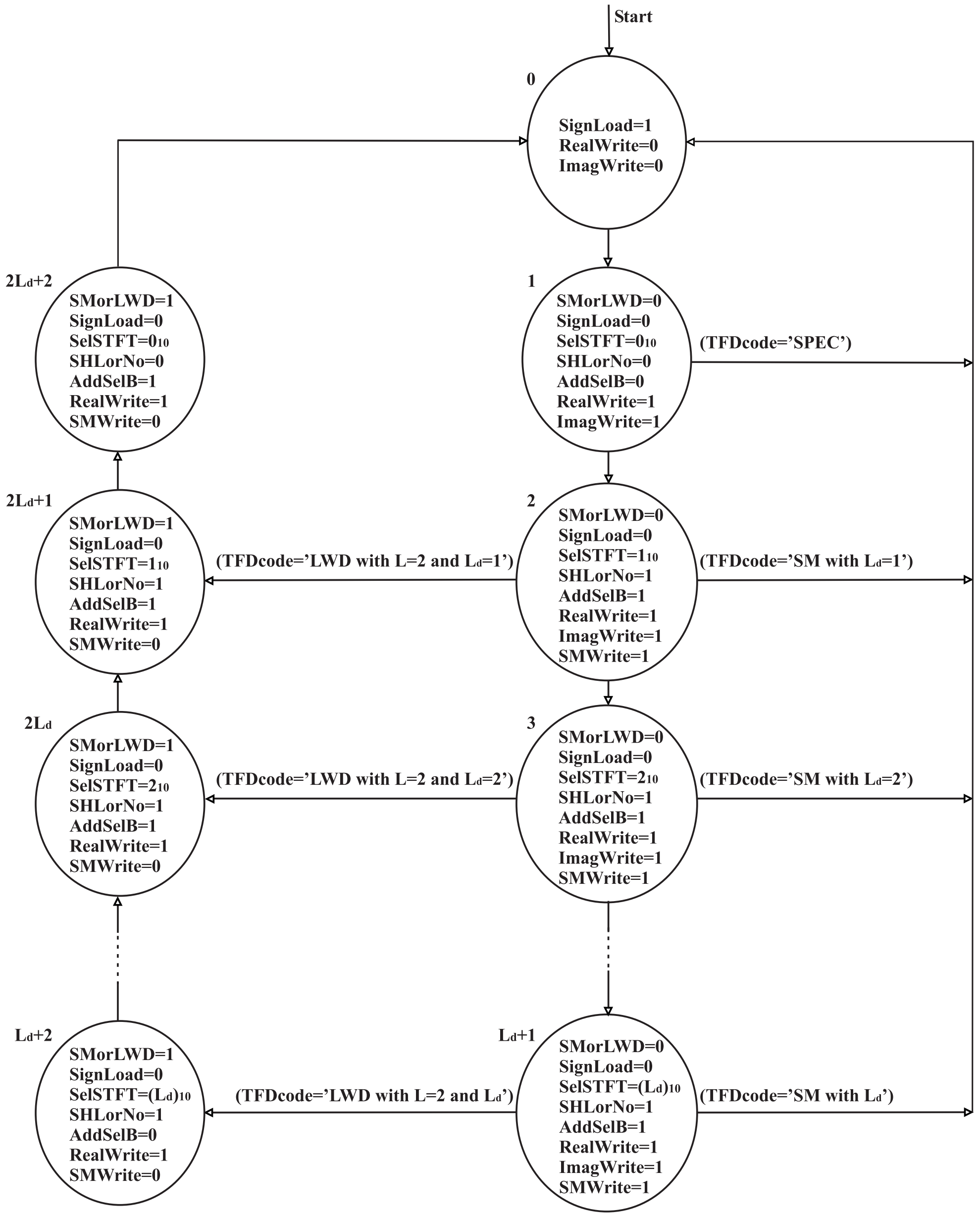

Figure 2: The finite state machine Control for the multicycle hardware implementation.

[5] K. Maharatna, A.S. Dhar, S. Banerjee: "A VLSI array architecture for realization of DFT, DHT, DCT and DST," *Signal Processing*, vol.81, no.9, Sept.2001, pp.1813-1822.

[6] L.L. Scharf, B. Friedlander: "Toeplitz and Hankel kernels for estimating time-varying spectra of discrete-time random processes," *IEEE Trans. SP*, vol.49, no.1, Jan.2001, pp.179-189.

[7] LJ. Stanković: "A method for time-frequency analysis," *IEEE Trans. SP*, vol.42, Jan.1994, pp.225-229.

[8] LJ. Stanković: "A method for improved energy concentration in the time-frequency signal analysis using the L-Wigner distribution", *IEEE Trans. SP*, vol.43, no.5, May 1995, pp.1262-1268.

[9] LJ. Stanković, V.N. Ivanović, Z. Petrović: "Unified approach to the noise analysis in the Wigner distribution and Spectrogram," *Ann. des Telecomm.*, vol.51, no.11-12, Nov./Dec.1996, pp.585-594.

[10] S. Stanković, V.N. Ivanović, LJ. Stanković: "Hartley transform based architecture for time-frequency analysis and time-varying filtering," *EUSIPCO*, Toulouse (France), Sept.2002, Vol.II, pp.217-220.

[11] 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. des Telecomm.*, vol.57, no.9-10, Sep./Oct.2002, pp.974-995.