Proceedings of the 5th WSEAS Int. Conf. on SIGNAL, SPEECH and IMAGE PROCESSING, Corfu, Greece, August 17-19, 2005 (pp44-49)

The VLSI Design of the Minimal Dimension, Cost and Power Consumption System for Time-Frequency Signal Analysis

VESELIN N. IVANOVIĆ, RADOVAN STOJANOVIĆ
Department of Electrical Engineering
University of Montenegro
Cetinjski put bb, 81000 Podgorica
MONTENEGRO
http://www.tfsa.cg.yu

Abstract: - Multiple clock cycle implementation (MCI) of a flexible system for time-frequency (TF) signal analysis is presented. Some very important and frequently used time-frequency distributions (TFDs) can be realized by using the proposed architecture: the spectrogram (SPEC) and the pseudo Wigner distribution (WD), as the oldest and the most important tools used in TF signal analysis; the S-method (SM) with various convolution window widths, as intesively used reduced interference TFD. This architecture is based on the short-time Fourier transformation (STFT) realization in the first clock cycle. It allows the mentioned TFDs to take different numbers of clock cycles and to share functional units within their execution. These abilities represent the major advantages of MCI design and they help reduce both hardware complexity and cost, as well as the power consumption reduction. In order to verify the results on real devices, proposed architecture has been implemented with a FPGA chips.

Key-Words: - Time-frequency analysis, S-method, Hardware realization, Multiple clock cycle implementation, FPGA devices.

1 Introduction and Problem Formulation

S-method (SM) for TF analysis alleviates serious drawbacks of the commonly used TFDs, spectrogram (SPEC) and WD: low concentration in the TF plane and generation of cross-terms in the case of multicomponent signals analysis, respectively, [7]. The SM definition is, [7]-[10]:

$$SM(n,k)=\sum_{i=-L_d}^{L_d} P_{(n,k)}(i)\times \times STFT(n,k+i)STFT^{*}(n,k-i) \tag{1}$$

where $STFT(n,k)$ represents the STFT of the analyzed signal $f(n)$, $2L_d+1$ is the width of a finite frequency domain (convolution) rectangular window $P(i)$, $P(i)=0$, for $|i|>L_d$ and the signal's duration is $N=2^m$. Definition (1), based on STFT, makes the SM 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. Some simple implementations of the systems for TF analysis are presented in [1]-[3], [9]-[12]. 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.

SM produces, as its marginal cases, the WD and the SPEC with maximal ($L_d=N/2$), and minimal ($L_d=0$) convolution window width, respectively. In the case of a multicomponent signal with non-overlapping components, by an appropriate window $P(i)$ width selection the SM can produce a sum of the WDs of individual signal components, avoiding cross-terms, [7]-[9]: $P(i)$ should be wide enough to enable complete integration over the auto-terms, but narrower than the distance between two auto-terms. In addition, the SM produces better results than the SPEC and the WD, regarding calculation complexity and noise influence, [7,8].

In order to involve only real multiplications in (1), we modify it by using $STFT(n,k)=STFT_{\mathrm{Re}}(n,k)+j\,STFT_{\mathrm{Im}}(n,k)$ ($STFT_{\mathrm{Re}}(n,k)$, $STFT_{\mathrm{Im}}(n,k)$ are the real and imaginary part of $STFT(n,k)$, respectively), as:

$$\begin{aligned}SM(n,k)&=STFT_{\mathrm{Re}}^2(n,k)+STFT_{\mathrm{Im}}^2(n,k)\\&+2\sum_{i=1}^{L_d}STFT_{\mathrm{Re}}(n,k+i)STFT_{\mathrm{Re}}(n,k-i)\\&+2\sum_{i=1}^{L_d}STFT_{\mathrm{Im}}(n,k+i)STFT_{\mathrm{Im}}(n,k-i).\end{aligned} \tag{2}$$

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

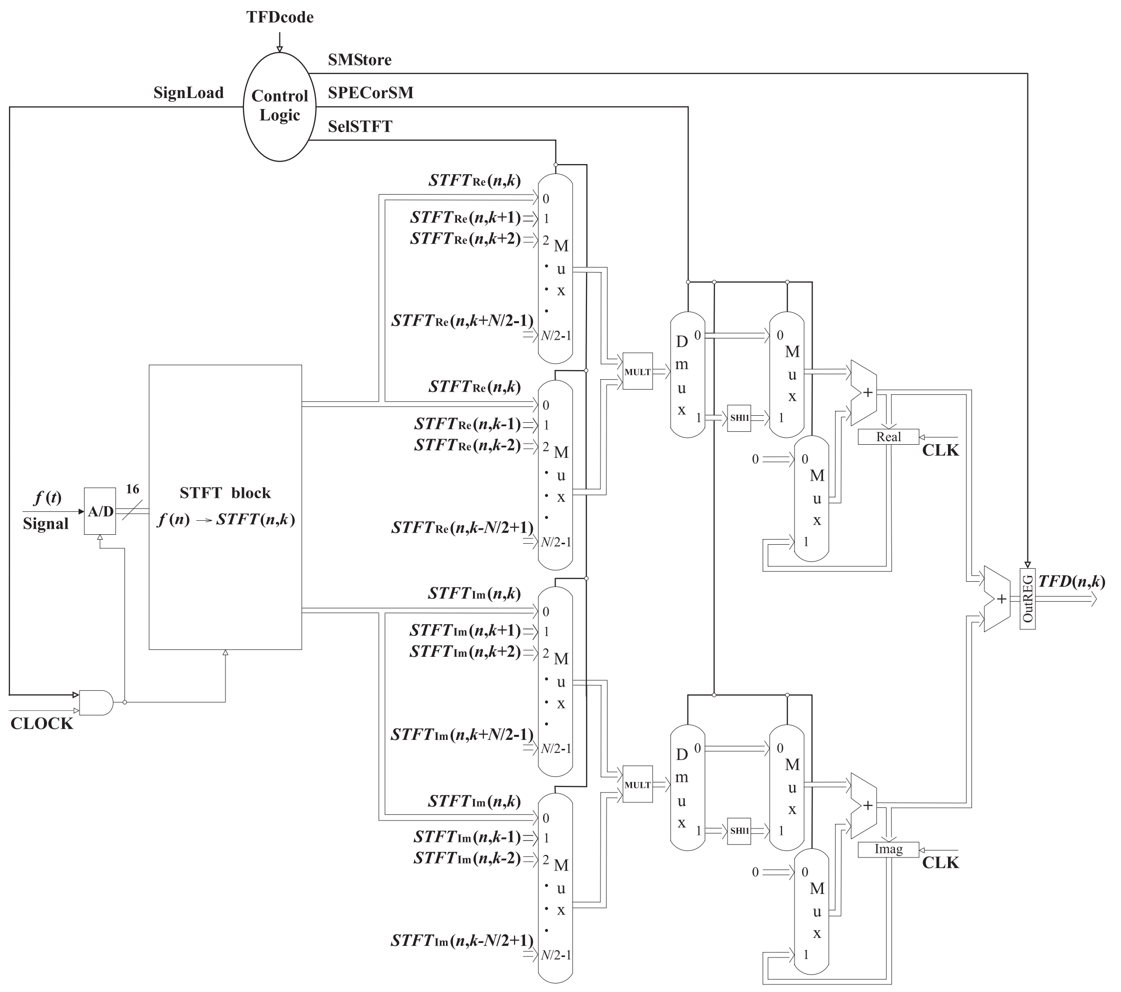

Fig. 1. Architecture for the multicycle implementation of the S-method for TF signal analysis.

2 Multicycle Implementation

The hardware necessary for one channel multicycle implementation of the SM is presented in Fig.1. Proposed hardware is designed based on a 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. The STFT block can be implemented by using the available FFT chips, or by using approaches based on the recursive algorithm, [1,2,9,10,12]. The second block is designed so that it realizes each summation term from the eq.(2) in the corresponding step of the method implementation.

We break the SM execution into several steps, each taking one clock cycle. Our goal in breaking the execution into clock cycles should be to balance the amount of work done in each cycle, so that we minimize the clock cycle time. In the first step, the STFT is calculated, in the second step the SPEC is calculated based on the first step execution. With each further step one realizes the SM with the incremental value of convolution window, based on the preceding steps. This improves the TFD concentration, aiming to achieve the one obtained by the WD.

Each sub-channel of the second block contains exactly one adder, one multiplier, and one shift left register for implementation of eq.(2). Because we need to use these functional units with different inputs in later steps, we must save the computed values, based on eq.(2), into a temporary registers that we name *Real* and *Imag*, respectively. In order to share functional units from the second block for different inputs in different clock cycles, we need to add multiplexors and/or a demultiplexor at their inputs. In the first step only the STFT block of the proposed two-block architecture is used, whereas in the other steps only the second block is used. This will be regulated by the set of control signals introduced on temporary registers, and multiplexors and a demultiplexor, see Table I.

By introducing the temporary registers and several multiplexors at the inputs of the functional units, we achieve the required reduction of the amount of hardware compared to a single-cycle architecture, [5,9,10] and Table II. Note that the achieved hardware reduction is significant, and it in-

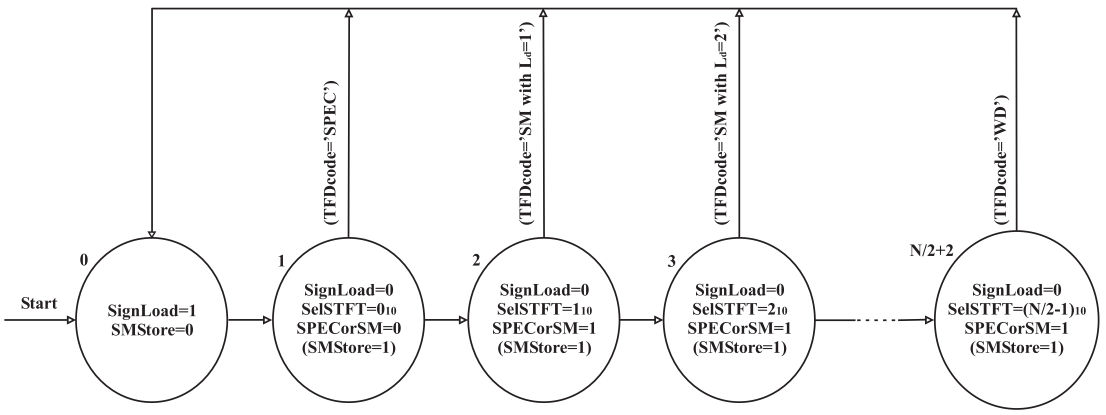

Fig.2. The finite state machine control for the architecture shown in Fig1. Output (SMStore=1) means that the *SMStore* control signal is asserted during only the final step of the corresponding TFD execution.

Table I. The function of each of the control signal generated by the Control logic.

Control signal	Effect
SelSTFT	(m-1)-bit signal which controls N/2-input multiplexors (two of them per subchannel are introduced to select between the STFT values from different channels.
SPECorSM	1-bit signal with the following functions: 1. Enables use of the Shift Left register in the corresponding steps (when we need to implement multiplication by 2), or disables this (in the second step); 2. Enables use of only one adder per sub-channel for implementing sums in eq.(2) by controlling its second input, which can be either the constant 0 (in the second step) or a register *Real* (or *Imag*) value (in each higher order step).
SignLoad	1-bit signal which enables sampling of the analyzed analog signal $f(t)$, but only after execution of the desired TFD of the analyzed signal samples from the preceding time instant.
SMStore	1-bit write control signal of the *OutREG* temporary register. It should be asserted during the step in which the SM with corresponding convolution window width is computed.

Table II. Total number of functional units per channel in an SM block and the clock cycle time in the cases of: a) single-cycle implementation (SCI) and b) the MCI. T_m is the multiplication time of a two-input 16-bit multiplier, T_a is the addition time of a two-input 16-bit adder, whereas T_s is the time for 1-bit shift.

Implementation	Adders	Multipliers	Shift Left Registers	Clock cycle time
SCI	$2L_d+1$	$2(L_d+1)$	$4L_d$	$2T_m+(L_d+3)T_a+T_s$
MCI	3	2	2	$T_m+2T_a+T_s$

creases as convolution window width increases. Since temporary registers and the introduced multiplexors are fairly small, this could yield a substantial reduction in the hardware cost, as well as in the used chip dimensions. Also, the reduced power consumption is achieved, which is strongly proportional to the chip capacity. Finally, the ability to realize almost all commonly used TFDs by the same hardware represents a major advantage of the proposed MCI design.

Defining the Control. From the defined multi-step sequence of the SM MCI execution we can determine what the Control logic must do at each clock cycle. It can set all based solely on the distribution code (TFDcode). Here, we use finite state Moore machine to specify the MCI control, Fig.2. Finite state Control essentially corresponds to the steps of SM execution; each state in the finite state machine will take one clock cycle.

3 Practical Implementation Approach

In this Section, the MCI architecture is implemented in the FPGA chips following the approach proposed in preceding section. The design was carried out in Altera Max+plus II software. For hardware realization the Altera's FLEX 10K chips family has been chosen. This family is fabricated in CMOS SRAM technology, running up to 100MHz and consuming less than 0.5mA on 5V. It has a high density of 10,000 to 250,000 typical gates, up to 40,960 RAM bits, 2,048 bits per embedded array block, and so on. The computation units are realized by using standard digital components in form of schematics entries or by AHDL (Altera Hardware Design Language) based mega-functions (Library of Parametrized Modules - LPM).

The FPGA-based implementation of the MCI architecture follows the design logic given in Fig.3.

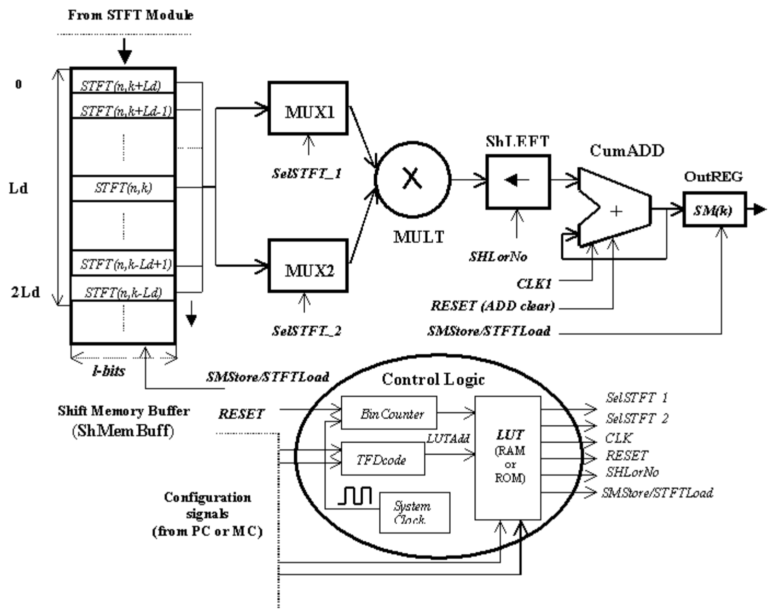

Fig.3. Block diagram of FPGA implementation of the MCI approach.

Since the real and imaginary computation lines are identical, the interpretation will be done through real ones. The STFT sample is imported from the STFT module to the *Shift Memory Buffer* (ShMemBuff) that is implemented as an array of parallel-in-parallel-out registers, Fig.4. Their outputs represent the STFT samples in time order *STFT*(n,k+L_d), *STFT*(n,k+L_d-1), ..., *STFT*(n,k), ..., *STFT*(n,k-L_d+1), *STFT*(n,k-L_d) and due to each *SMStore/STFTLoad* cycle they have been shifted for one position. These are also fed to the inputs of multiplexors MUX1 and MUX2 and, two-by-two, regarding on multiplexor's addresses *SelSTFT*_1 and *SelSTFT*_2, forwarded to the parallel multiplier MULT in order to produce partial product term according to the eq.(2). This term is either shifted left or not, depending on the signal *SHLorNo*. The cumulative pipelined adder *CumADD* has been designed to replace an adder and a multiplexor (addressed by the *AddSelB* control signal) from Fig.1. The multiplying and shifting operations are parallel, while the adding has a latency of one clock. After L_d+1 clocks, the output of the *CumADD* will contain the sum *SM*(n,k) that represents the final value of the SM. The next two cycles can be used for the signals *SMStore/STFTLoad* and RESET that will store the sum *SM*(n,k) in the output register and reset *CumADD* to zero, respectively. Use of the RESET signal will increase the calculation time for one clock. It means that the calculation process takes L_d+3 cycles, one more than is elaborated in Fig.2. Note that the RESET signal can be generated by the signal *SMStore/STFTLoad*, using a short delay, that will reduce the calculation process to L_d+2 cycles. In order to clarify the principle of calculation and simulation (the process of cumulative sums *cumSM* represented in Fig.5), we have used the first variant of RESET generation, with L_d+3 clocks.

Look-Up-Table (LUT), realized in the form of ROM or RAM memory and represented in Table III, manages the computation process. The binary counter (see Fig.3) generates the low LUT's addresses, while *TFDcode* register sets the high ones. It means that starting address of the running memory block is assigned to the corresponding value L_d stored in *TFDcode* register. At the end of the sequence, the binary counter is cleared by the signal RESET. During system initialization, the memory's contents and value of *TFDcode* register are automatically loaded from outside, by using PC or general-purpose microcontroller. Of course, these parameters can be permanently stored using ROMs, EEPROMs and FLASHs instead of RAMs.

Fig.4 shows a schematic diagram for SM calculation from the STFT samples (STFT to SM gateway) using MCI approach. It is implemented for $L_d \leq 3$ and N=8. The control logic is realized by using ROM. The maximal register widths for each unit determine the capacity of the assigned chip. The critical point is the width of the *CumADD*, which can be represented as a function of both STFT data length l and the maximal possible convolution window width $L_{d\max}$ that can be implemented by using proposed architecture,

$$\mathrm{CEIL}(\log_2((2^{2l+1}-1)\cdot(L_{d\max}+1))) . \qquad (3)$$

In order to verify the chip operation before its programming, the compilation and simulation have been performed by using the various test vectors. An example of simulation is shown in Fig.5.

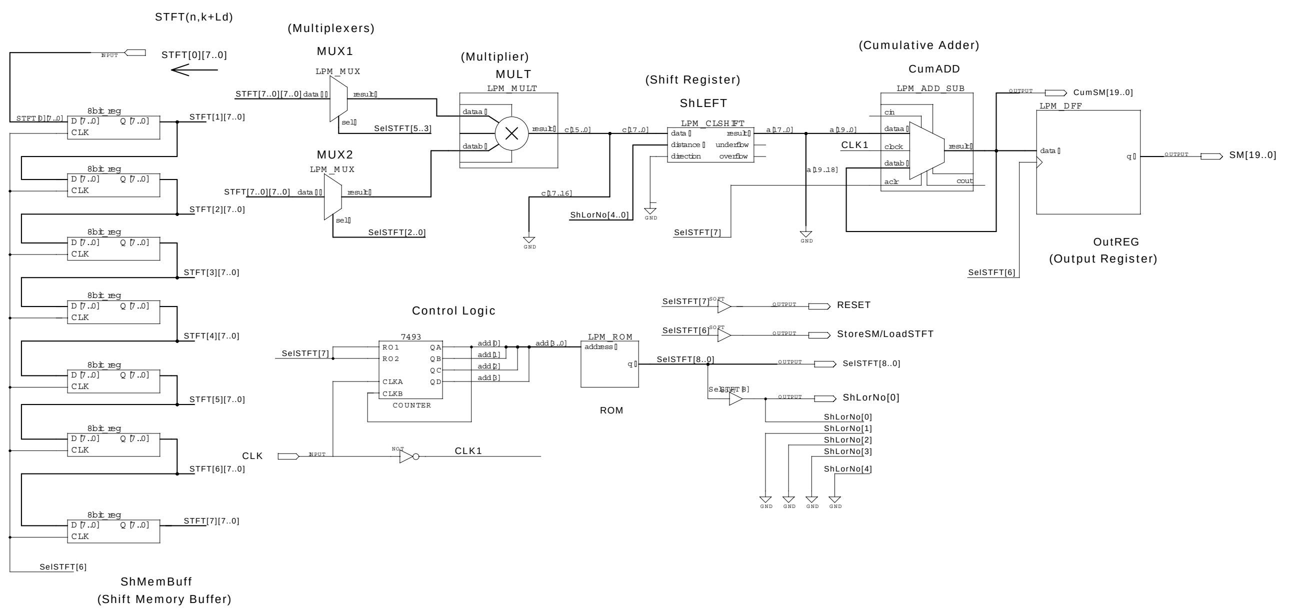

Fig.4. The schematic diagram of the 8-bit STFT to SM gateway implemented in FPGA using MCI approach.

Table III. LUT's values for given L_d. The $\mathrm{ADD}(STFT(n,k))$ means the address location of sample $STFT(n,k)$ inside *ShMemBuff*, whereas $m = \mathrm{CEIL}(\log_2 N) = \mathrm{Length}(SelSTFT\_1)$. Simbol "<<" denotes logical shifl left operation.

LUT's memory locations	Control signals area			MUXs' addresses	
	SHLorNo	RESET	*SMStore/STFTLoad*	*SelSTFT_1* bits	*SelSTFT_2* bits
0	0	0	0	$\mathrm{ADD}(STFT(n,k)) << m$	$\mathrm{ADD}(STFT(n,k))$
1	1	0	0	$\mathrm{ADD}(STFT(n,k+1)) << m$	$\mathrm{ADD}(STFT(n,k-1))$
…	1	0	0	…	…
L_d	1	0	0	$\mathrm{ADD}(STFT(n,k+L_d)) << m$	$\mathrm{ADD}(STFT(n,k-L_d)$
L_d+1	0	0	1	0	0
L_d+2	0	1	0	0	0

During the test phase we have implemented 8-bit and 16-bit computation configurations for MCI architecture. Having in mind the design symmetry, both real and imaginary parts have been developed separately or together. Some implementation details for L_d=3, N=8 and selected real devices from 10K family are summarized in Table IV. Advantageous of the represented results of the MCI design as compared with the SCI ones against usual criteria such as chip capacity, computation speed, power consumption and cost are given in [5].

4 Conclusion

Flexible system for TF signal analysis is proposed. Its MCI design is presented. Proposed architecture can be used for real-time implementation of some commonly used TFDs. It allows a functional unit to be used more than once per TFDs execution, as long as it is used on different clock cycles, and, conesquently, enables a signifycant reduction of hardware complexity and cost. The major advantages of the proposed design are the ability to allow implemented TFDs to take different numbers of clock cycles and to share functional units within a TFDs execution. Finally, proposed architecture is practically verified by its implementations in FPGA devices.

References:

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

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

[3] B. Boashash, and J.B. Black, "An efficient real time implementation of the Wigner-Ville distribution," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. 35, no. 11, 1987, pp. 1611-1618.

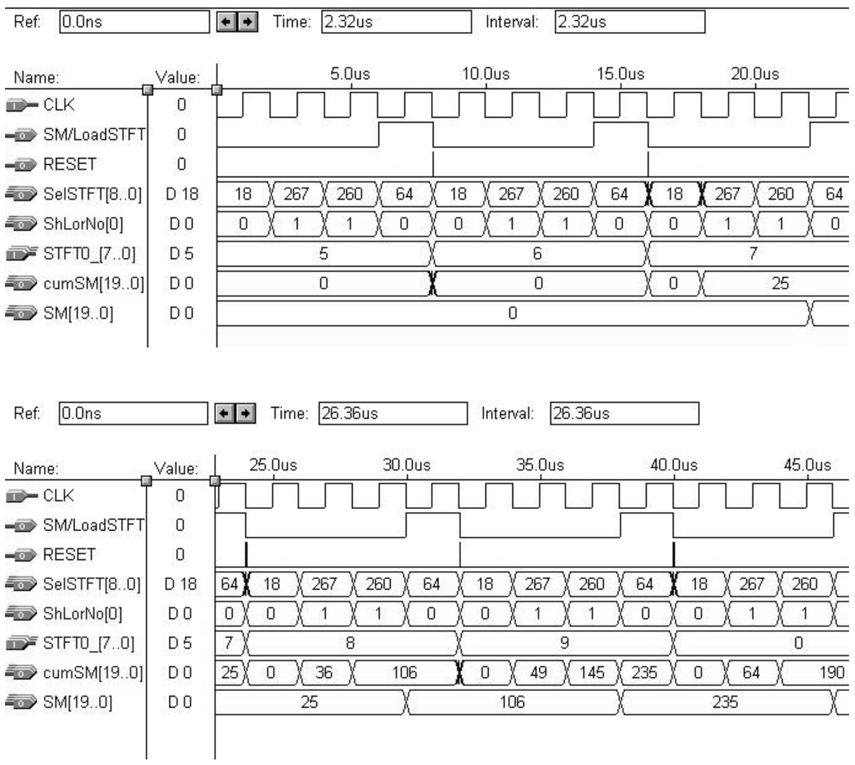

Fig.5. Simulation illustration for test vector V={5,6,7,8,9,0,0,…} and L_d=3.

Table IV. Summarized implementation utilization for real devices and L_d=3, N=8, and l=8 and l=16.

Computation Architecture	Total Logic Cells (LCs) used	Total Flip-Flops used	Memory Bits used	Total I/O Pins used	Utilized LCs for recommended device	Recommended device
Real_8-bits MCI	641	101	144	41	55%	EPF10K20TC144-3
Real_16-bits MCI	1772	197	144	69	76%	EPF10K40RC208-3
Real+Imag_8-bits MCI	1281	198	144	69	74%	EPF10K30RC208-3
Real+Imag_16-bits MCI	3543	397	144	125	94%	EPF10K70RC248-3

[4] V.N. Ivanović, and LJ. Stanković, "Multiple clock cycle real-time implementation of a system for time-frequency analysis," in *Proc. of the 12th EUSIPCO*, Viena, Austria, Sept.2004, pp. 1633-1636.

[5] V.N. Ivanović, R. Stojanović, and LJ. Stanković, "Multiple clock cycle architecture for the VLSI design of a system for time-frequency analysis," *EURASIP Journal on Applied Signal Processing, Special issue on Design methods for DSP systems*, in print.

[6] D.A. Patterson, and J.L. Hennessy, *Computer organization and design, the hardware/softwa-re interface*, Morgan Kaufmann Publishers, San Mateo, California, 1994.

[7] LJ. Stanković, "A method for time-frequency analysis," *IEEE Trans. SP*, vol. 42, no. 1, 1994, pp. 225-229.

[8] LJ. Stanković, V.N. Ivanović, and Z. Petrović, "Unified approach to the noise analysis in the Wigner distribution and spectrogram," *Ann. Telecomm.*, vol. 51, no. 11-12, 1996, pp. 585-594.

[9] S. Stanković, and 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.

[10] S. Stanković, LJ. Stanković, V.N. Ivanović, and R. Stojanović, "An architecture for the VLSI design of systems for time-frequency analysis and time-varyin filtering," *Ann. Telecomm.*, vol. 57, no. 9-10, 2002, pp. 974-995.

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

[12] K.J.R. Liu, and C.T. Chiu, "Unified parallel lattice structures for time-recursive discrete cosine/sine/Hartley transforms," *IEEE Trans. SP*, vol. 41, no. 3, 1993, pp. 1357-1377.