

An economical modified VLSI architecture for computing power spectral density supported welch method

S. Kalvikkarasi^{a,*} and S. Saira Banu^b

^a*Karpagam University, Coimbatore, Tamil Nada, India*

^b*Department of Electronics and Communication Systems, Karpagam University, Coimbatore, Tamil Nada, India*

Abstract. The Welch algorithm furnishes a good estimate of the spectral power at the expense of high computational complexity. The primary intension is to compute the FFT of the individual non-overlapped parts (i.e., half of the original segments) and acquire the FFT of the overlapped segments by merging those of the non-overlapped segments. In this paper, initially the input discrete signal is subjected to an $L/2$ -point FFT and then the two successive segments are merged to L -point segment using a modified architecture utilizing an improved Fractional Delay Filter(FDF) design by adapting a Multiplier less implementation for efficient contribution. The merged segments are then subjected to a window filter, designed using delay lines and shifters replacing the multiplier blocks. Finally the power spectral density (PSD) is computed by computing the periodogram and then averaging the periodogram for the windowed segments. Complete module is realized using Xilinx_ISE software with the target device as xc4vfx100-12-ff1152. The design is coded in verilog HDL. The functional verification of the proposed design reported a PSD with an error of 5.87% when compared with the similar Matlab PSD computation. The synthesis results confirm the efficiency and computational complexity reduction of the proposed architecture when comparing with similar existing researches.

Keywords: Welch algorithm, fractional delay filter, periodogram, power spectral density, window filter

1. Introduction

Power spectral estimation of periodic and random signals may be considered as one among the most significant application domains in digital signal processing (DSP). Spectrum analysis serves as the primary step in speech recognition problems for achieving reduction in the speech bandwidth and to process the acoustic data. More complicated spectrum analyses are carried out in SONAR systems for finding out the submarines as well as the surface vessels. The spectral measurements made in radar are helpful in locating the target and to acquire the information regarding velocity, though the measurements involved in spectrum

analysis are boundless. In signal processing, spectral analysis is extensively employed to differentiate and to track the required signals [1,2] For instance, the analysis of radar and sonar signals [3], spectrum sensing [4,5] and information retrieval as in biomedical signal analysis [6].

Power spectral density estimation involves two important methods, namely, the non-parametric methods and the parametric methods [8]. Non-parametric methods are utilized, if the knowledge about the signal before a particular instant of time is less. While comparing with the parametric methods, the computational complexity associated with the non-parametric methods is found to be very small. The non-parametric methods can be again classified into periodograms and correlograms. Periodograms are occasionally called as direct methods because they produce direct data transformation. The sample spectrum, Bartlett's method, Welch's method and the Daniell Periodogram come

*Corresponding author: S. Kalvikkarasi, Karpagam University, Coimbatore, Tamil Nada, India. E-mail: kalvikkarasis0677@gmail.com.

under the category of periodogram methods. Correlograms, on the other hand, make use of Wiener-Khinchin theorem [7] and hence, termed as indirect methods. In all correlogram based methods, Fourier transform is applied on the estimate of the autocorrelation sequence. Utilization of fewer data samples in correlation has resulted in large variance that is related to higher order lags and as a result, windowing becomes necessary.

Usually, the parametric approaches suffer from high computational complexity. The signals are applied with Fourier transform in the periodogram based approaches. Several investigations on Fourier transform can be found in the literature. The presence of computationally-efficient Fourier transforms has caused the periodograms to be more used than the other parametric methods. A familiar non-parametric, periodogram-based method that is helpful in estimating the power spectral density is the Welch PSD method [9]. A number of researches have been made for designing short-time Fourier transform, rather than designing architectures for PSD computation [10,11]. A systolic architecture, which relies on the ARMA model, has been presented in [3]. Here, the design of the architectures was direct and novelty in optimizations was not found. The Welch method is a modified periodogram approach that is extensively used for computing the PSD [9]. The key component in Welch method is the FFT. In a usual sense, for dividing the input signal into numerous segments, an overlap of 50% is utilized. This means, for two successive FFT operations, half the samples remain unaltered.

The aim of this work is to propose new changes in the Welch PSD method, so that a low-complexity architecture that is appropriate for low-power embedded systems can be obtained. One such application is the analysis of biomedical signal, in which a devoted hardware can be utilized in systems with reduced cost and power. Several parameters like area, performance and the amount of power consumed has to be considered, if PSD computation has to be included into the biomedical monitoring systems because these type of systems demand severe power consumption constraints. The main contribution of our work includes (a) A modified merging process with a multiplier less computation for converting two $L/2$ -point FFTs into a single L -point FFT (b) A multiplier less Hanning window architecture and (c) a modified periodogram computation process.

The rest of the sections in this paper are organized as follows: Section 2 discusses about the recent research works that are related to the proposed work. The in-

spiration as well as the methods involved in the proposed work is suggested in Section 3. Various figures and equations that are related to the proposed work are dealt in Section 4. Finally, Section 5 provides the results of experimentation, their explanation and comparative analysis with the other existing techniques.

2. Related works

The FFTs consume very large power that major part of the energy can be saved, if the FFT blocks are left unused. Depending on various radix algorithms, several FFT architectures have been presented in the literature. The radix-2 algorithm only decides the energy consumed by the FFT. Modifications in these estimates occur with the radix of the algorithm and the architecture that lie beneath. Optimizations in FFT architecture can be accomplished to handle real signals and currently, several FFT architectures that handle real-valued signals have been proposed. In [12], the first pipelined architecture that performs RFFT computation depending on the Cooley-Tukey algorithm has been introduced. With this pipelined architecture, less number of operations can be achieved by solving the irregularities in RFFT. This method can be used in both the decimation in time (DIT) as well as the decimation in frequency (DIF) decompositions and has the ability to be generalized for any number of points N with power of 2. In addition, this pipelined circuit was capable of performing reordering and hence, the problems associated with the output order of the samples can be rectified. In [13], a generalized approach for designing efficient architectures for the computation of RFFT has been suggested. This approach can be extended to radix-2 5 and higher radix algorithms. Especially, novel 2-parallel and 4-parallel pipelined architectures have been created depending on the modified flow graph and hybrid datapath design with radix-2 3 and radix-2 4 algorithms. The optimization in the data paths were achieved with the folding methodology. While comparing this architecture with the other existing architectures, various advantages that include low hardware complexity, less number of adders and delays can be achieved because real butterflies insist the usage of half the number of adders and real data paths necessitate half the number of storage elements.

Practically, the adjustable FDF is required to be effectively realized. Nearly all the papers have dealt with the problem of designing the Farrow structure for accomplishing reduction in the number of structural

multiplications, adders and delay elements. In [14], a novel scheme for FDF has been proposed to implement the sub-filters in the Farrow structure in an efficient way. When comparing with other prior techniques in literature, reduction in arithmetic complexity can be achieved. But, this becomes impossible with long word length of filter coefficient. Considerable register savings can be produced with this new architecture, so that the rise in rare arithmetic can be overcome. In [15], a system of steerable parametric loudspeaker was designed and a complete solution has been proposed, encompassing all steps from signal acquisition to primary wave emissions. FD filter was used here for removing the limitation of delay interval. The utilization of Farrow structure FIR filter has produced fractional delay for the design of steerable parametric loudspeaker. With the Fractional Delay filter, some sort of arbitrary delays can be produced for controlling the steering angle of the beam in a precise manner. Field Programmable Gate Array (FPGA) serves better for the system of steerable parametric loudspeaker because it offers improved parallel and real-time processing. The FD filter with Farrow structure was used in FPGA for implementing variable steering angles in a continuous manner.

In [16], DA-based formulation of BLMS algorithm was proposed. In this algorithm, both the convolution operation for computing the filter output and the correlation operation for computing the weight-increment term could be carried out through the same LUT. The presented DA-based design makes use of a new look-up table (LUT)-sharing technique, to compute the filter outputs and weight-increment terms of BLMS algorithm. Reduction in the number of LUT words to be updated per output reduces external logic and the amount of power consumed. The work presented in [17] include a parallel FIR architecture that take advantage of the inherent nature of symmetric coefficients for decreasing half the number of multipliers in the sub filter section by increasing the adders in preprocessing and post processing blocks. Use of adders instead of multipliers is beneficial because adders consume reduced silicon area. In addition, the use of adders in preprocessing and post processing blocks does not increase with increase in the length of the FIR filter. On the other hand, the number of multipliers multiplies with the length of the FIR filter.

In [18], a low-complexity scheme for obtaining the power spectral density through Welch method has been proposed. The computational complexity gets lessened, but degradation in performance has occurred

with this scheme. The developed approach computes the even samples accurately and a fractional-delay filter helps in estimating the odd samples that cause a minute error. The difference in the spectral power obtained through the proposed and original methods is around 8% with a 4-tap bidirectional FDFilter. A new architecture has been proposed depending on the modified method. With the proposed architecture that uses a 4-tap bidirectional FD filter, 33% less energy has been consumed than the original method.

3. Research methodology

It can be noted from the existing works that the core components include FFT, FFT Estimator, Window filter and absolute-square multiple accumulator (AMAC) circuits. The FFT estimator block contains a butterfly structure and a FD FIR filter. The window filter is a 3-tap symmetric FIR filter. The AMAC circuit is used for computing the periodograms and to average them over M segments. It is evident from the analysis that the overhead in the computational complexity, power consumption and area depends on the FFT estimator and the windowing filter block. Hence, the aim of this work is to design and implement the hardware architecture for power spectral density, by making changes in FDF and the Window Filter architecture. Multipliers were not employed in the implementation of the two sub modules, so that the complexity in the computation with increased area and power consumption can be alleviated. Implementation of the proposed PSD computation along with the entire changes is performed with verilog language in Xilinx-ISE. The synthesis and implementation tool in Xilinx-ISE helps in generating the logic utilization report and the xpower analyzer tool allows the power analysis report to be generated. These reports are then compared against those of the earlier techniques.

4. Proposed method

The proposed VLSI architecture for Power spectral Density estimation using the well known Welch algorithm with reduction in power consumption and area is described in this section. The main intension of our design is to reduce the computation complexity, power and area of the hardware by modifying the architecture of original Welch power spectral architecture. Studies shows that half of the samples are the same over two

consecutive FFT operations and this property can be used for the reduction in the number of operations required to compute the PSD. Basic PSD computation using Welch method includes the following steps,

Step.1: The input discrete signal $\varphi(l)$ is split up into M segments of length L , overlapping by ρ points. In most of the cases $\rho = L/2$, the overlap is said to be 50%.

$$\varphi_m(l) = \varphi(l + (M - 1)\rho) \quad (1)$$

Where, $l = 0, 1, \dots, L - 1$ and $m = 1, 2, \dots, M$.

Step.2: Once the signal is split up into overlapping segments, suitable window function is applied individually for the M -segments.

$$\varphi_{Wm} = \varphi_m(l) \times W \quad (2)$$

Step.3: FFT process is then applied to each of the windowed segments separately.

$$\Phi_m(k) = \sum_{l=0}^{L-1} \varphi_{Wm}(l) \cdot e^{-j2\pi lk/L} \quad (3)$$

Step.4: A Modified periodogram process is performed with each of the segments.

$$P_{m(k)} = \frac{1}{L} |\Phi_m(k)|^2 \quad (4)$$

Step.5: Finally the PSD is computed by averaging the periodograms of the M -segments.

$$PSD = \frac{1}{M} \sum_{m=0}^M P_{m(k)} \quad (5)$$

In our modified computational process for power spectral density include the following steps.

Step.1: The input discrete signal $x(n)$ is split up into $M + 1$ - segments with non-overlapping points of length $L/2$. In our cases $\rho = 0$, hence there is no overlap between two successive segments.

Step.2: $L/2$ -point FFT process is then performed for each segment separately.

Step.3: The merging of two $L/2$ -point FFTs into a single L -point FFT is then carried out by adopting a modified merging process with a multiplier less computational unit, which is one of the contributions in this research.

Step.4: Windowing in frequency domain is then performed for each L -point FFT using Hanning window coefficients; designed using a 3-tap FIR filter without the use of multipliers, which is another important contribution in this work.

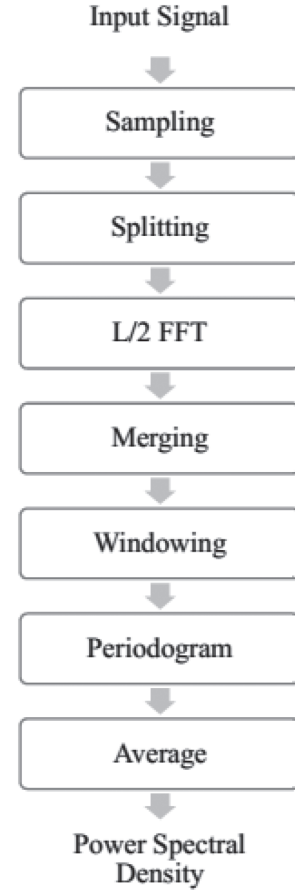


Fig. 1. Proposed power spectral density estimation process.

Step.5: A Modified periodogram process is performed for each of the windowed L -point FFT.

Step.6: Finally the PSD is computed by averaging the periodograms of the M -segments.

The proposed process for power spectral density computation is shown in Fig. 1.

4.1. PSD computation architecture

The block schematic architecture for our proposed PSD Computation unit adopting the modified Welch PSD estimation technique is shown in Fig. 2.

The modified architecture consists of a $L/2$ -point FFT-processor, which process the input discrete signals. A buffer with $L/2$ -samples size is connected with the FFT processor for storing the FFT outputs of previous segments. For example if the FFT processor is processing M^{th} segment of the input signal, then the FFT output of the $M + 1^{\text{th}}$ segment is stored in the buffer for the merging process. L -point modified merg-

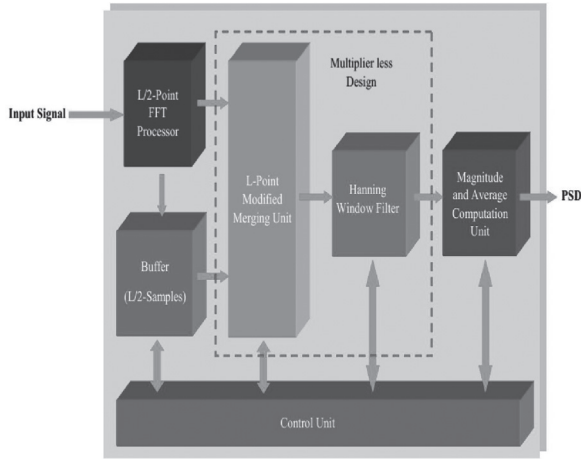


Fig. 2. Proposed Welch PSD estimation architecture.

ing unit in our architecture is used to merge two $L/2$ samples from M^{th} and $M + 1^{\text{th}}$ segment to a single L -point segment with low complexity computations by adopting the Discrete Architecture technique for replacing the multiplier with just some precomputed values stored in memory, a shifter and an accumulator. Once the $L/2$ samples are merged to L samples, then the samples are filtered out by a Hanning window, realized in hardware using a 3-tap FIR filter. The FIR filter is redesigned in such a way that the filtering operation is carried out without using any multiplier operation by adopting shifters and adders. The output from the FIR filter is then fed to a magnitude and Average computation unit, which functions similar to the latest designs available in the literatures, adopting fewer resources. Our contribution for this research can be briefed as below,

4.1.1. *L-point modified merging unit.*

From [18], the merging of two consecutive segments with fewer computations can be represented as,

For Even Sample,

$$S_{\text{Even}} = X_{\alpha}(\rho) + X_{\beta}(\rho) \tag{6}$$

For Odd Sample,

$$S_{\text{Odd}} = X_{\alpha}(\rho + 0.5) + X_{\beta}(\rho + 0.5) \tag{7}$$

From the above relation, for computing L -point merged segment the sample S_{Even} can be computed just by adding two samples from both the M^{th} and $M + 1^{\text{th}}$ segments. This can be realized by adopting an adder, but in case of computing S_{Odd} a fractional delay of 0.5

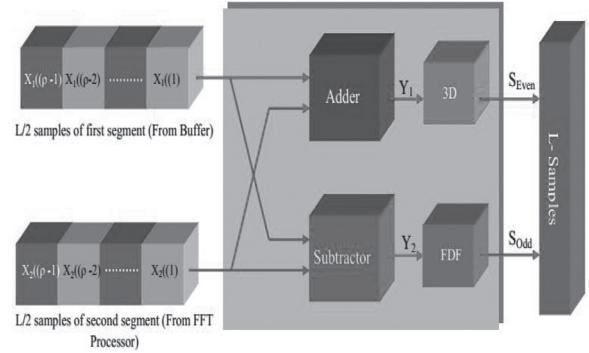


Fig. 3. Architecture for modified merging unit.

(half sample delay) is required in addition to a subtraction. In practical the realization of a fractional delay of $Z^{\frac{1}{2}}$ is non-causal, since it requires the sampling of a future signal to a half delay. For making it a casual system we have to design an efficient Fractional Delay filter with suitable pipelines. The hardware architecture for our modified Merging unit is as shown in Fig. 3. Here two segments are considered for merging. The addition and subtraction of two samples can be realized using a butterfly unit structure, since both the operations require the similar samples. The even sample is then computed with 3 delay lines and the odd sample is computed using a fractional delay filter.

4.1.2. *Fractional Delay Filter design*

A Fractional Delay Filter is a device for band limited interpolation between samples. It has wide range of applications in fields of signal processing, including communications, speech processing, array processing and music technology. For these fractional delay filter applications, the general concern is interpolating an input signal's values approximately between two sampling points by adopting varying mathematical techniques. The fractional delay D is a fraction of a sample point, which is a real value between 0 and 1. In terms of ideal interpolating application, Shannon's signal reconstruction formula may be used from the sampling theorem in order to reconstruct the continuous-time signal from acquired samples. The reconstruction formula is given as,

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT) \text{sinc} \left[\frac{\omega_s}{2} (t - nT) \right] \tag{8}$$

Where, $x(t)$ is a continuous-time signal, T is the sampling interval and ω_s is 2π multiplied by the sampling frequency. Examining the reconstruction formula

in Eq. (8), the ideal interpolator has an impulse response of,

$$h_c(t) = \text{sinc}\left(\frac{\omega_s t}{2\pi}\right) \quad (9)$$

The formula given in Eq. (9) helps in the determination of a reconstructed continuous-time signal from the discrete-time sampled input signal. In order for the above conversion formula to be consistent for the fractional delay application, the input sampled signal must be delayed by the desired delay D , where,

$$D = D_{int} + d \quad (10)$$

D_{int} refers to an integer delay, and d refers to the fraction delay somewhere around 0 and 1. Acquire the delayed interpolated discrete time output, Eqs (9) and (10) are considered for a reconstructed discrete time signal that is shifted and re-sampled by the important delay parameter D in Eq. (11)

$$y(n) = x(n - D) = \sum_{k=-\infty}^{\infty} x(k) \text{sinc}(n - D - k) \quad (11)$$

As to Eqs (9) and (11), the impulse response of this ideal system is obtained by shifting and sampling the infinitely long Sinc function, which yields a noncausal system. In order to obtain the best approximation result, the desired total fractional delay D of Eq. (11) should be between the two central taps of an FIR filter for odd ordered filters or within half a sample from the central tap for even ordered filters. As a result, the delay should follow the inequality

$$\frac{L-1}{2} \leq D \leq \frac{L+1}{2} \quad (12)$$

Where L is the order of the filter. The integer portion of the delay D_{int} should follow the equation for odd ordered filters:

$$D_{int} = \frac{L-1}{2} \quad (13)$$

For even ordered filters

$$D_{int} = \left\{ \begin{array}{l} \frac{L}{2}, 0 \leq d < \frac{1}{2} \\ \frac{L-1}{2}, \frac{1}{2} \leq d < 1 \end{array} \right. \quad (14)$$

Several design methods exist for finite impulse response fractional-delay filters. The least-squared integral error design approach is considered in this work.

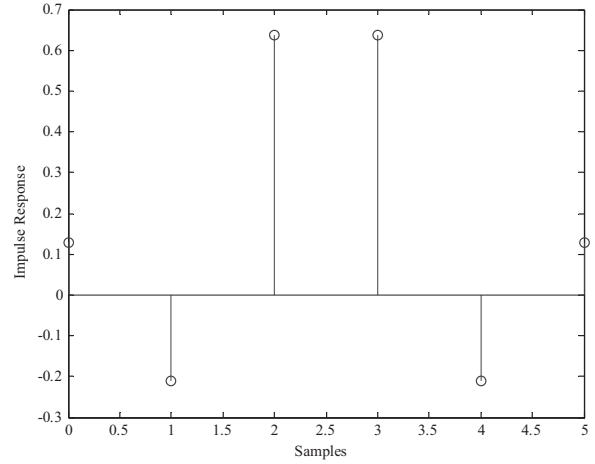


Fig. 4. Impulse response for FDF.

The impulse response of an L^{th} order least-square FD FIR filter can be expressed as,

$$h(l) = \begin{cases} \text{sinc}(l - 0.5 - \frac{l}{2} + 1), & 0 \leq l \leq L-1 \\ 0, & \text{Otherwise} \end{cases} \quad (15)$$

From previous studies, a filter of length $L = 6$ can be adapted for our PSD computation process. The impulse response is given by $h(0) = 0.1273$, $h(1) = -0.2122$, $h(2) = 0.6366$, $h(3) = 0.6366$, $h(4) = -0.2122$ and $h(5) = 0.1273$. The main source of complexity in computation, power consumption, and area overhead is from the multiplication block, which is the most important block in designing a FIR filter. In our design DA based architecture is adopted for replacing the multiplier block in FIR filter for filter coefficient multiplication. The modified architecture for fractional delay using 6-tap FIR filter is shown in figure. From Eq. (15), the impulse response for our fractional delay filter architecture is as shown in Fig. 4.

For fractional delay computation in our method, past and future samples are required to compute the output. In order to make available of the future and past samples, in our architecture we have added 4 delay lines. The values $b1n$, $b2n$ and $b3n$ are the sum of corresponding future sample and the past sample respectively. In conventional FIR filter computation process after computing the sum each value is subjected to a coefficient multiplication process followed by a summation process. In our case we have six coefficient values hence the computation requires 6 constant multipliers which contribute much in the computation complexity with added raise in power and area of the com-

plete architecture. Since the coefficients exhibit symmetrical property, the adder and constant multiplier can be reduced to a half. Even though by adapting this property reduces the number of multiplier count, there are still 3 multipliers. For effective contribution, all the three constant multiplier blocks have to removed, since multipliers add more complexity in computation and occupies major power and area resources.

In order to remove the usage of multiplier blocks, the function of the multiplier is carried out by a DA based computation technique. Area savings from using DA can be up to 80% in DSP hardware designs. The advantage of a distributed arithmetic approach is its efficiency of mechanization. This approach employs no explicit multipliers in the design, only Look-up tables (LUTs), shift registers, and a scaling accumulator. All of these functions efficiently map to FPGAs. Distributed Arithmetic is introduced into the design of FIR filters as follows.

$$y = \sum_{k=1}^k C_k x_k \tag{16}$$

Where, $x_k \Rightarrow \{bk_0, bk_1, \dots, bk(N-1)\}$ – N bit scaled two's complement number; bk_0 – sign bit.; C_k – constant value; x_k can be expressed as,

$$x_k = -b_{k0} + \sum_{n=0}^{N-1} b_{kn} 2^{-n} \tag{17}$$

Substituting Eq. (17) in Eq. (16), we get,

$$y = - \sum_{k=1}^k (C_k \cdot b_{k0}) \sum_{k=1}^k \left[\sum_{n=0}^{N-1} C_k \cdot b_{kn} 2^{-n} \right] \tag{18}$$

The hardware realization of the above Eq. (18) can be done as shown in Fig. 5. The LUT stores all possible partial products over the filter coefficient space. The LUT contents are tabulated in Table 1. When the LUT is subjected with any of the possible combination as shown in the table, the corresponding pre-computed value stored in the LUT is output to the succeeding blocks in the architecture shown in Fig. 5.

The inputs to the LUT are fed from a FIFO register bitwise. Once the input is fed, the corresponding values stored is output, which in turn is shifted and added with the previous output value and finally when the odd sample is output when the bit length reaches the MSB (Most Significant Bit). Hence the fractional delay filter incorporated within the modified merging unit can be designed without utilizing the multiplier block.

Table 1
Possible input combinations and the corresponding stored values

b1n	b2n	b3n	Stored values
0	0	0	0
0	0	1	0.6366
0	1	0	-0.2122
0	1	1	0.4244
1	0	0	0.1273
1	0	1	0.7639
1	1	0	-0.0849
1	1	1	0.5517

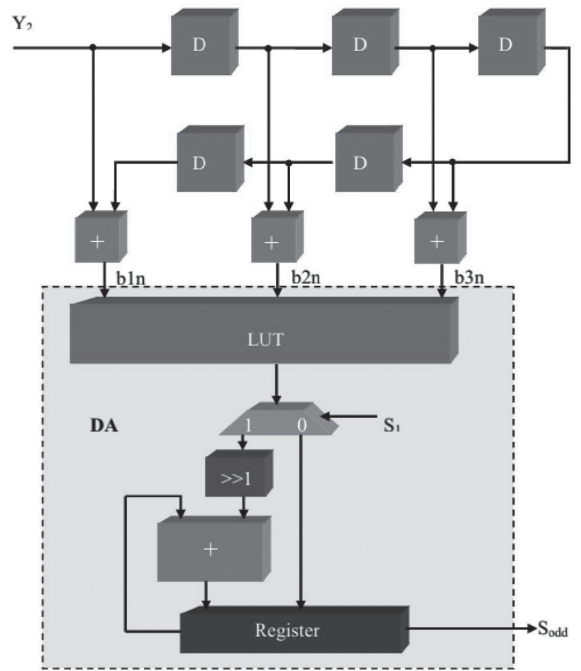


Fig. 5. DA based FDF.

4.1.3. Window filter design

Since the windowing operation has to be performed in the frequency domain the multiplication operation in the time domain has to be replaced with a convolution operation. Windowing functions are most easily understood in the time domain; however, they are often implemented in the frequency domain instead. Mathematically there is no difference when the windowing is implemented in the frequency or time domains, though the mathematical procedure is somewhat different. The convolution operation is computationally complex compared to simple multiplication operation. However, the hamming window function used in the PSD computation typically represent raised cosine functions and these can be represented by 3 non-zero coefficients while the rest of them are close to zero. Further the filter is symmetric, i.e., two of the fil-

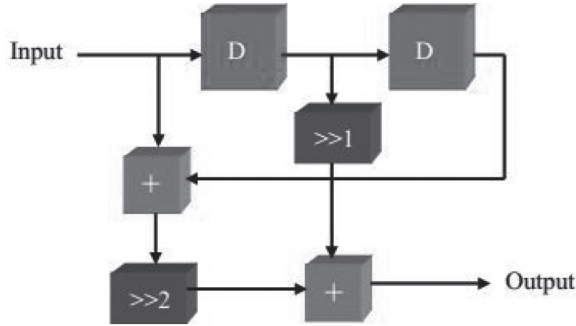


Fig. 6. Multiplier less window filter architecture.

ter coefficients are equal. Therefore, in general the convolution operation can be implemented using 2 multiplication and 2 addition operations. But the implementation of hamming window filter still needs 2 multiplication blocks for hardware realization. Comparative analysis of the coefficients of Hamming window filter with Hanning window filter reveals that the coefficients are $h_{Ham} = \{-0.23, 0.54, -0.23\}$ and $h_{Han} = \{-0.25, 0.5, -0.25\}$ respectively.

Comparing both the window filter coefficients, in hardware realization the coefficients of Hanning window is a better option for implementation, since the coefficient $0.5 = \frac{1}{2} = 2^{-1}$ can be implemented by just shifting right the input value by 1-bit. The coefficient $0.25 = \frac{1}{4} = 2^{-2}$ can be implemented by using a 2-bit right shifter. In hardware realization the convolution operation can be implemented using a 3-tapFIR filter in the frequency domain. The 3-tapFIR filter architecture with the above described modifications is shown in Fig. 6.

The hardware realization of hanning window is based on the following equations,

Let the inputs be samples $s_1, s_2, s_3 \dots, s_{L-1}$

Where, L is the window size, here $L = 1024$.

For each input sample the corresponding windowed sample is given as,

$$W_{sn} = 0.5(s_n + s_{n-2}) + 0.25(s_{n-1}) \quad (19)$$

Where, W_{sn} – Current windowed output sample; s_n – Current input sample; s_{n-1}, s_{n-2} – Previous input samples.

4.1.4. Magnitude and average computation unit

The magnitude of each sample in the M^{th} segment can be computed by first removing the negative sign of sample if found and then squaring each values. The absolute magnitude of each sample thus computed is

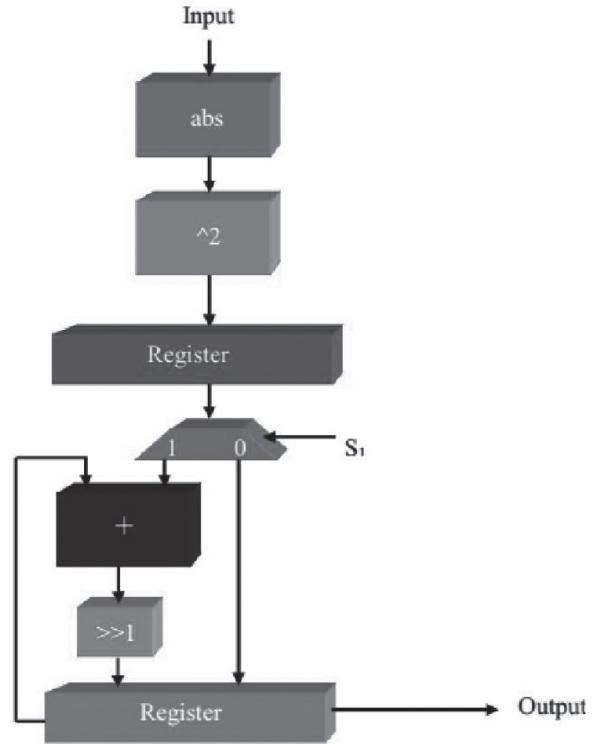


Fig. 7. Architecture of magnitude and average computation unit.

then stored inside a register that can store L -samples and this gives the periodogram for each segment. The Averaging of periodogram for the M -segments is done by shifting right the sum of the previous accumulated value with the present value. This can be represented as,

$$\begin{aligned} \frac{P_1 + P_2 + P_3 + \dots + P_M}{M} &= \frac{P_1 + P_2}{2} + \\ &\frac{(P_1 + P_2) + P_3}{2} \dots + \\ &\frac{(P_1 + P_2 + \dots + P_{M-1}) + P_M}{2} \end{aligned} \quad (20)$$

Where, $P_1, P_2, P_3 \dots, P_M$ are the periodogram of the respective segments.

5. Results and discussion

To evaluate the resource usage, Power consumption and speed of the proposed VLSI Architecture for Computing Power Spectral Density, several synthesis experiments were performed. The complete module is coded in Verilog HDL. All synthesis results were obtained for the Xilinx Virtex -4 FPGA (xc4vfx100-12-

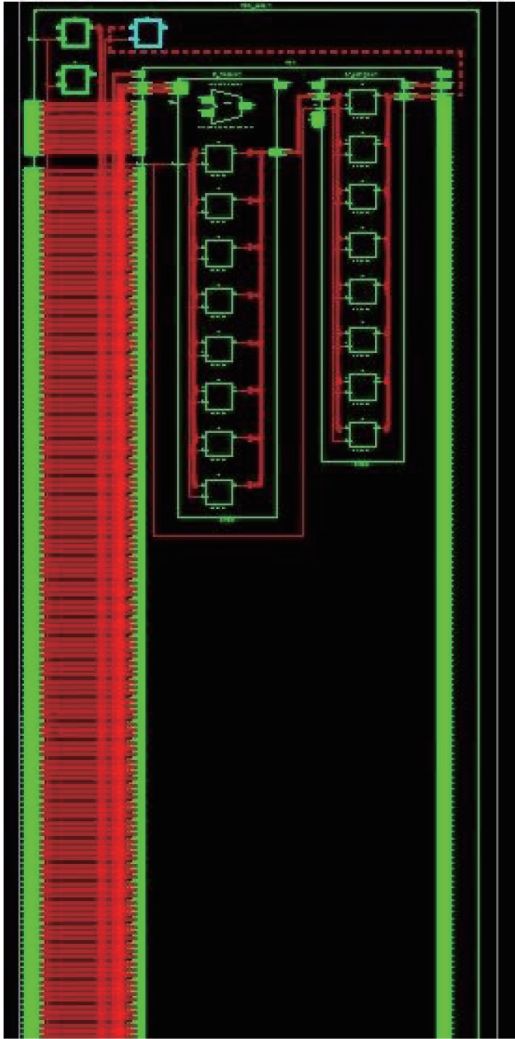


Fig. 8. RTL Schematic of Our PSD computation architecture.

ff1152) after place & route using Xilinx ISE v14.1. Matlab and Modelsim are used as the simulation platforms. We can analysis the changes between the input signal and the output signal to observe the permanence of the designed PSD architecture through Matlab, while observing the real-time implementation performance of FPGA through Modelsim.

The RTL schematic for our proposed PSD computation architecture is as shown in Fig. 8.

5.1. Area

The device Utilization report of the complete module is tabulated in Fig. 9. Among the available 42176, 4902 slices is utilized. In the target device 84352 counts of Flip flops and 4 Input LUTs are present

(Since each slice includes 2 flipflops and LUTs) and among this only 5% of the former and 10% of the latter are utilized.

Figure 10 tabulates the resource utilized by our modified merging unit. Only 19 out of 10,944 slices, 14 out of 10944 Flip Flops and 25 out of 10944 4 input LUTs are the occupied resources by our architecture.

The above Fig. 11 organizes the asset used by our implemented Hanning Window filter. 79 out of 10,944 Flipflops, 86 out of 10944 LUTs and 76 out of 10944 4 Slices are the possessed assets by our Hanning Window filter.

5.2. Power

Power consumed by our proposed architecture is analyzed using Xpower analyzer tool in the Xilinx software. The input supply voltage is $V_{ccint} = 1.2v$, $V_{ccaux} = 2.500v$ and $V_{cco25} = 2.5v$.

In Fig. 12 the corresponding current flow increases as the clock frequency increases towards the maximum frequency of the system. Figures 13 and 14 represents the current flow with respect to the voltages $V_{ccaux} = 2.500v$ and $V_{cco25} = 2.5v$.

The power consumption to the system increases with respect to the increase in frequency. The static power remains the change and is mainly because of the leakage and the dynamic power increases in a very small measure as the frequency increases towards the maximum frequency and the relation is plotted in Fig. 15 below.

5.3. Performance

To observe the performance of the implementation, a suitable EEG signal (shown in Fig. 16) is precomputed using the matlab and then the sampled signal is fed to the Xilinx tool in the form of text file. The output from the implemented module is then saved as text file which is then read in the matlab and the graph is plotted. Meanwhile the same input is fed to a matlab script that performs similar to the proposed technique and the graph is plotted. For comparison the PSD computed using our implemented module and the similar approach implemented using Matlab is shown in Figs 17 and 18. Since the FDF design approximates the samples, there exists an error of 5.87% between PSD outputs of Matlab and Xilinx. The precomputed EEG signal consists of 4608 samples and each segment consists of 512 samples.

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	4902	42176	11%
Number of Slice Flip Flops	4560	84352	5%
Number of 4 input LUTs	9120	84352	10%
Number of bonded IOBs	16257	576	2822%
Number of GCLKs	1	32	3%

Fig. 9. Device Utilization summary of the complete module.

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	14	10,944	1%
Number of 4 input LUTs	25	10,944	1%
Number of occupied Slices	19	5,472	1%
Number of Slices containing only related logic	19	19	100%
Number of Slices containing unrelated logic	0	19	0%
Total Number of 4 input LUTs	28	10,944	1%
Number used as logic	24		
Number used as a route-thru	3		
Number used as Shift registers	1		
Number of bonded IOBs	65	240	27%
IOB Flip Flops	16		
Number of BUFG/BUFGCTRLs	1	32	3%
Number used as BUFGs	1		
Average Fanout of Non-Clock Nets	1.25		

Fig. 10. Device Utilization summary of our modified merging unit.

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	79	10,944	1%
Number of 4 input LUTs	86	10,944	1%
Number of occupied Slices	76	5,472	1%
Number of Slices containing only related logic	76	76	100%
Number of Slices containing unrelated logic	0	76	0%
Total Number of 4 input LUTs	134	10,944	1%
Number used as logic	86		
Number used as a route-thru	48		
Number of bonded IOBs	48	240	20%
Number of BUFG/BUFGCTRLs	1	32	3%
Number used as BUFGs	1		
Average Fanout of Non-Clock Nets	2.01		

Fig. 11. Device Utilization summary of the Hanning Window filter.

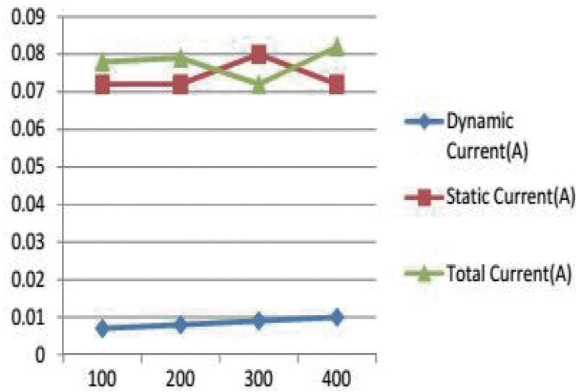


Fig. 12. Current flow with respect to change in frequency.

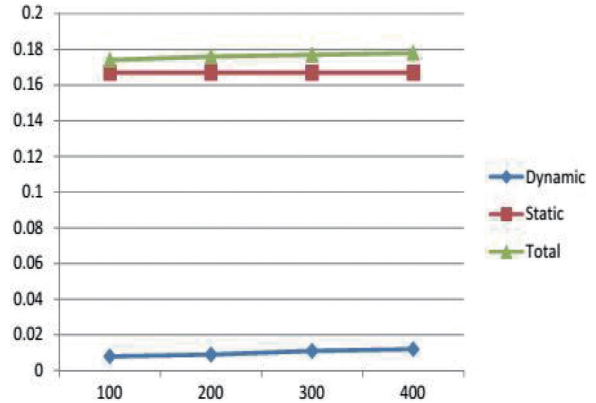


Fig. 15. Power consumption with respect to frequency.

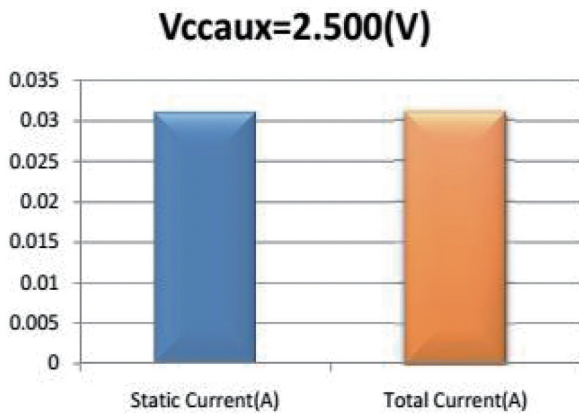


Fig. 13. Current flow when input Vccaux = 2.500v.

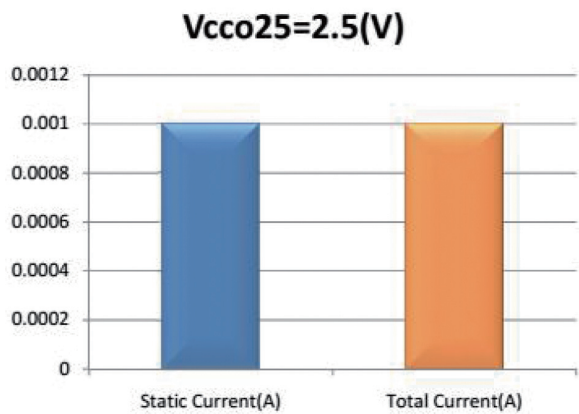


Fig. 14. Current flow when input Vcco = 2.500v.

5.4. Complexity analysis

The complexity in computation is based on the number of multiplications and number of addition operations required for the complete computation process.

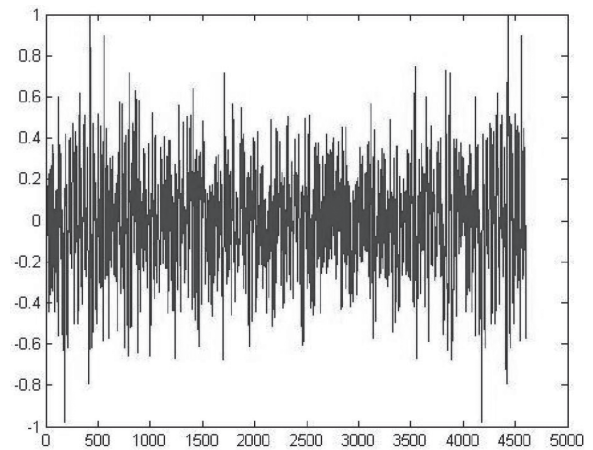


Fig. 16. Input EEG signal.

In our case since the Magnitude and Average Computation Unit include computations more or less similar to the existing and the original Welch method, for the complexity computation it is not considered. The number of additions and multiplication required for the conventional Welch method, the work proposed in [18] and our PSD computation technique is given as,

The FFT process considered here for complexity analysis is a basic radix-2 algorithm. Since our merging unit and window filter design does not require any multiplication operation the complexity is reduced much comparing to other existing methods. Also the number of additions given here is exception of the DA process, since the DA can be implemented using the latest technique and the adder requirement can be much reduced. In case, if the addition operation range a little high due to the DA process this will not affect the complexity in computation much since the complex multiplication operations in completely removed

Table 2
Computational complexity comparison

	Addition	Multiplication
Original	$L \log L$	$\frac{L}{2} \log L + L$
[18]	$\frac{L}{2} \log(\frac{L}{2}) + 3L + \frac{L}{2}(L' - 1)$	$\frac{L}{4} \log(\frac{L}{2}) + 2L + \frac{L}{2} \frac{L'}{2}$
Our method	$\frac{L}{2} \log(\frac{L}{2}) + 3L + \frac{L}{2}(L' - 3)$	$\frac{L}{4} \log(\frac{L}{2})$

Table 3
Computational complexity comparison with various input signal length L

L	Multiplications			Additions		
	Original	Keshab K. Parhi and Manohar Ayinala	Proposed Method	Original	Keshab K. Parhi and Manohar Ayinala	Proposed Method
1024	6144	5120 (16.66%)	2304 (62.5%/55%)	10240	10240 (0%)	9216 (10%/10%)
2048	13312	10240 (23.07%)	5120 (61.53%/50%)	22528	21504 (4.54%)	19456 (13.63%/9.52%)
4096	28672	20992 (26.78%)	11264 (60.71%/46.34%)	49152	45056 (8.33%)	40960 (16.66%/8.33%)

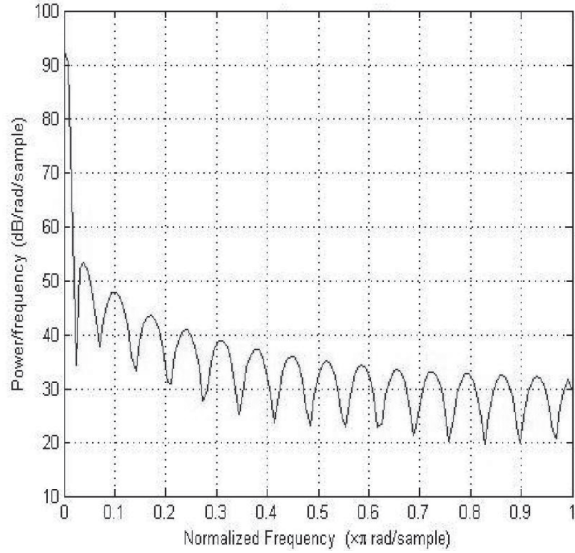


Fig. 17. PSD computed using Matlab.

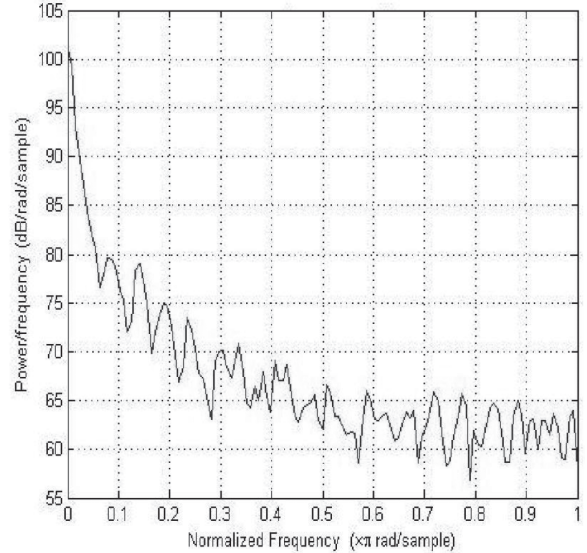


Fig. 18. PSD computed using Xilinx.

in our technique. The comparison of number of multiplication and additions for various length of input signal L is tabulated in Table 3. In Table 3, L represent the FFT length and L' represent the length of the fractional delay filter. The computations involved in computing periodogram and averaging remains the same in all the three approaches and hence they are excluded from the results. From the comparison it is evident that our approach requires less number of computations (in terms of addition and multiplication) than that of conventional Welch approach and the work proposed in [18].

From the comparison it is obvious that, In terms of multiplication operation our proposed technique exhibit 61.5% and 50.4% savings comparing with

the original Welch technique and the work proposed in [18] respectively and a savings of 13.4% and 9.28% in terms of addition operation is also exhibited by our proposed technique.

6. Conclusion

It is observed that the advantages of the proposed approach are limited to applications where errors introduced due to short filters are acceptable. Biomedical signal processing application is one such application where a machine learning algorithm is tolerant to errors (noise) introduced using the proposed approach.

An efficient modified VLSI architecture for computing power spectral density based on Welch Method was proposed in this paper. The complexity in computation, hardware utilization and power consumption was reduced much in this work by modifying the merging and window filter blocks of PSD architecture with suitable multiplier less operations. The complete architecture is realized using verilog HDL in Xilinx ISE and then synthesized and simulated using the same. The synthesized reports for area and power are tabulated and verified. The output obtained from simulation using an EEG signal input is then compared with the output from a Matlab script with the similar function as that of our proposed technique and found an error percentage of 5.87%. The complexity computation confirmed that our technique require about 55% reduction in multiplication and about 11% reduction in addition when comparing with similar existing works. In future, the proposed PSD architecture can be incorporated within a signal processing system like EEG analyzer for determining the depressive disorders in medical field.

References

- [1] P. Stoica and R. L. Moses, Introduction to Spectral Analysis. Englewood Cliffs, NJ, USA: Prentice-Hall, 1997.
- [2] M. Hayes, Statistical Digital Signal Processing and Modeling. New York, NY, USA: Wiley, 1996.
- [3] F. El-Hawary and T. Richards, A systolic computer architecture for spectrum analysis, Proc. OCEANS, vol. 4, pp. 1061–1065, Sep. 1989.
- [4] T.-H. Yu, C.-H. Yang, D. Cabric and D. Markovic, A 7.4 mW 200 MS/s wideband spectrum sensing digital baseband processor for cognitive radios, in Proc. Symp. VLSI Circuits, Jun. 2011, pp. 254–255.
- [5] T. Yucek and H. Arslan, A survey of spectrum sensing algorithms for cognitive radio applications, *IEEE Commun. Surveys Tutorials* **11**(1) (2009), 116–130.
- [6] Y. Park, L. Luo, K. K. Parhi and T. Netoff, Seizure prediction with spectral power of EEG using cost-sensitive support vector machines, *Epilepsia* **52**(10) (Oct. 2011), 1761–1770.
- [7] A. Oppenheim and R. Schaffer, Discrete Time Signal Processing, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall.
- [8] S. Haykin, Adaptive Filter Theory, 4th ed. Englewood Cliffs, NJ, USA: Prentice-Hall.
- [9] P. D. Welch, The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms, *IEEE Trans. Audio Electroacoustics*, vol. AU-15, pp. 70–73, Jun. 1967.
- [10] S. Zhang, D. Yu and S. Sheng, A discrete STFT processor for realtime spectrum analysis, in Proc. IEEE Asia Pacific Conf. Circuits Syst., 2006, pp. 1943–1946.
- [11] A. Sanchez, M. Garrido, L. Vallejo, J. Grajal and C. Lopez-Barrio, Digital channelised receivers on FPGAs platforms, in Proc. IEEE Int. Radar Conf., 2005, pp. 816–821.
- [12] M. Garrido, K.K. Parhi and J. Grajal, A pipelined FFT architecture for real-valued signals, *IEEE Trans. Circuits Syst. I, Reg. Papers* **56**(12) (Dec. 2009), 2634–2643.
- [13] M. Ayinala and K.K. Parhi, FFT architectures for real-valued signals based on radix-23 and radix-24 algorithms, *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, 2013.
- [14] Muhammad Abbas, Oscar Gustafsson and Håkan Johansson, On the Fixed-Point Implementation of Fractional-Delay Filters Based on the Farrow Structure, *IEEE Transactions On Circuits And Systems – I: Regular Papers*, Vol. 60(4), 926–937, April 2013.
- [15] S. Wu, M. Wu, C. Huang and J. Yang, FPGA-based implementation of steerable parametric loudspeaker using fractional delay filter, *Applied Acoustics* **73** (2012), 1271–1281.
- [16] B.K. Mohanty and P.K. Meher, A High-Performance Energy-Efficient Architecture for FIR Adaptive Filter Based on New Distributed Arithmetic Formulation of Block LMS Algorithm, *IEEE Transactions On Signal Processing* **61**(4) (15 February 2013).
- [17] Y.-C. Tsao and K. Choi, Area-Efficient VLSI Implementation for Parallel Linear-Phase FIR Digital Filters of Odd Length Based on Fast FIR Algorithm, *IEEE Transactions On Circuits And Systems – II: Express Briefs* **59**(6) (June 2012).
- [18] K.K. Parhi and M. Ayinala, Low-Complexity Welch Power Spectral Density Computation, *IEEE Transactions On Circuits And Systems – I: Regular Papers* **61**(1) (January 2014).

Copyright of International Journal of Knowledge Based Intelligent Engineering Systems is the property of IOS Press and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.