Middle-East Journal of Scientific Research
24 (Recent Innovations in Engineering, Technology, Management & Applications): 171-178, 2016
ISSN 1990-9233; © IDOSI Publications, 2016
DOI: 10.5829/idosi.mejsr.2016.24.RIETMA128

An Enhanced VLSI Architecture for Calculating Power Spectral Density Based on Welch Method

S. Kalvikkarasi and Saira Banu

Department of Electronics and Communication System,
Karpagam University, Coimbatore, Tamilnadu, 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 (Fast Fourier Transform) 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 (Power Spectral Density) with an error of 5.87% when compared with the similar Mat lab PSD computation. The synthesis results confirm the efficiency and computational complexity reduction of the proposed architecture when comparing with similar existing researches.

Key words: Welch algorithm · Fractional Delay Filter · Periodogram · Power Spectral Density · Window filter

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 direct methods because they produce direct data transformation. The sample spectrum, Bartlett's method, Welch's method and the Daniell Periodogram come 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.

Corresponding Author: S. Kalvikkarasi, Department of Electronics and Communication System, Karpagam University, Coimbatore, Tamilnadu, India.

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 the biomedical monitoring systems because these type of systems demand severe power consumption constraints.

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 show 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%.

$$\varphi_m(l) = \varphi\ (l + (M-1)\rho) \tag{1}$$

where, $l = 0, 1, \ldots\ldots, l-1$ and $m=1, 2, \ldots\ldots, M$.

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

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

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

$$\Phi m(k) = \sum_{l=0}^{L-1} \varphi_{Wm}(l) e^{-j2\pi lk/L} \tag{3}$$

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

$$P_{m(k)} = \frac{1}{L}\left|\Phi m(k)\right|^2 \tag{4}$$

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

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

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.

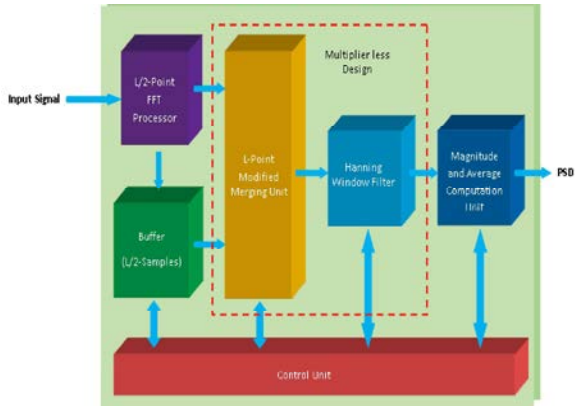


Fig. 1: Proposed Welch PSD estimation architecture

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.

PSD Computation Architecture: The block schematic architecture for our proposed PSD Computation unit adopting the modified Welch PSD estimation technique is shown in Figure 1

***L*-point Modified Merging Unit:** The merging of two consecutive segments with fewer computations can be represented as, For Even Sample,

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

For Odd Sample,

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

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 $M+1^{th}$ segments. This can be realized by adopting an adder, but in case of computing S_{Odd} a fractional delay of 0.5 (half sample delay) is required in addition to a subtraction. In practical the realization of a fractional delay of $\frac{1}{z^{\frac{1}{2}}}$ 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 Figure 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.

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,

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

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

$$h_c(t) = sinc(\frac{\omega_s t}{2\pi}) \tag{9}$$

The formula given in (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 consistant for the fractional delay application, the input sampled signal must be delayed by the desired delay *D*, where,

$$D = D_{\text{int}} + d \tag{10}$$

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, (9) and (10) are considered for a reconstructed discrete time signal that is shifted and re-sampled by the important delay parameter D in (11)

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

As to (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 (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

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

where L is the order of the filter. The integer portion of the delay Dshould follow the equation for odd ordered filters:

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

For even ordered filters

$$D_{\text{int}} = \begin{cases} \frac{L}{2}, & 0 \le d < \frac{1}{2} \\ \frac{L-1}{2}, & \frac{1}{2} \le d < 1 \end{cases} \quad (14)$$

Several design methods exist for finite impulse response fractional-delay filters. The least-squared integral error design approach is considered in this work. The impulse response of an L^{th} order least-square FD FIR filter can be expressed as,

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

From previous studies, a filter of length L=6 can adapted for our PSD computation process. The impulse response is given by

$$h(0) = 0.1273, \quad h(1) = -0.2122,$$
$$h(2) = 0.6366, \quad h(3) = 0.6366,$$
$$h(4) = -0.2122 \text{ and } h(5) = 0.1273$$

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

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 filter 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 andrespectively.

Comparing both the window filter coefficients, in hardware realization the coefficients of Hanning window is a better option for implementation, since the coefficient can be implemented by just shifting right the input value by 1-bit. The coefficient 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.

Magnitude and Average Computation Unit: The magnitude of each sample in the 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 then stored inside a register that can store -samples and this gives the periodogram for each segment. The Averaging of periodogram for the -segments is done by shifting right the sum of the previous accumulated value with the present value. This can be represented as,

$$\frac{P_1+P_2+P_3+.....+P_M}{M} = \frac{P_1+P_2}{2} + \frac{(P_1+P_2)+P_3}{2} ..... + \frac{(P_1+P_2+.....+P_{M-1})+P_M}{2} \quad (20)$$

where, are the periodogram of the respective segments.

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-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 Figure 2

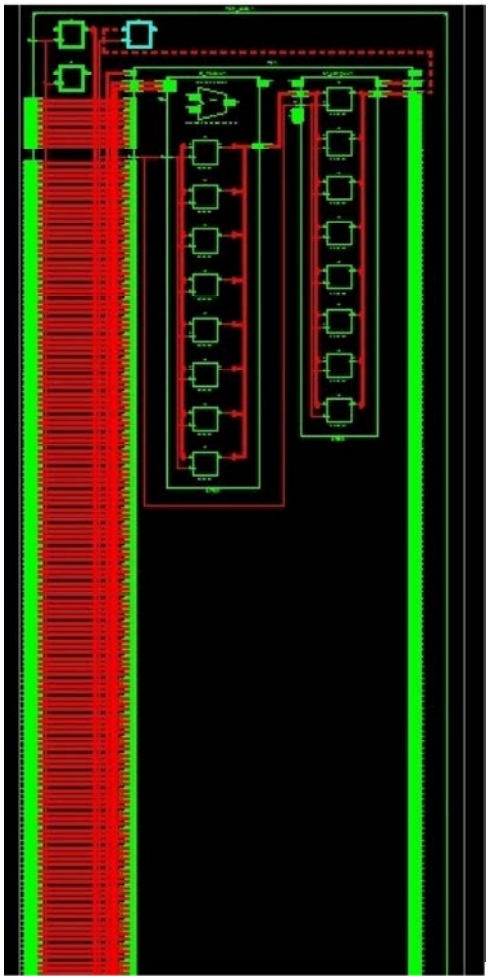


Fig. 2: RTL Schematic of Our PSD computation architecture

a)Area:

Table 1: Device Utilization summary of the complete module

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	4902	42176	11%
Number of Slice Flip Flops	4868	84352	5%
Number of 4 input LUTs	9128	84352	10%
Number of bonded IOBs	16357	576	2802%
Number of BUFG/BUFGCTRLs	1	32	3%

The device Utilization report of the complete module is tabulated in table 3. 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.

Table 2: Device Utilization summary of our modified merging unit

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		

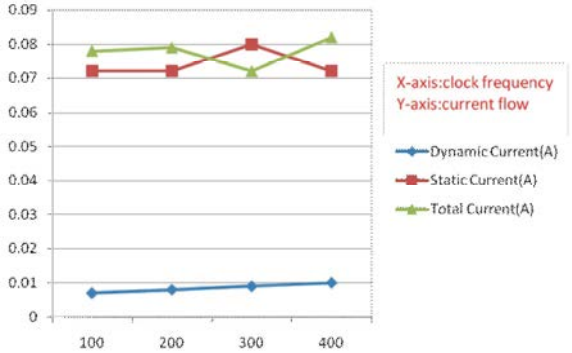
Table 2 tabulates the resource utilized by our modified merging unit. Only 19 out of 10,944 slices, 14 out of 10944 Flip Flops and 25 out of 10944 4 input LUTs are the occupied resources by our architecture.

Table 3: Device Utilization summary of the Hanning Window filter

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		

The above table 3 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 b)Power:

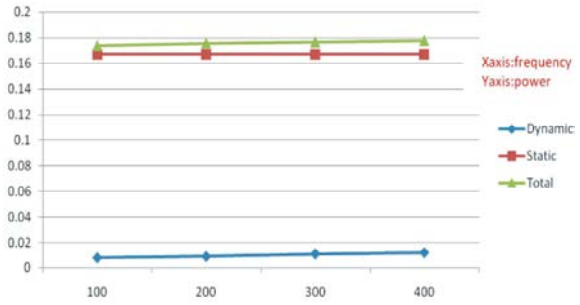
Power consumed by our proposed architecture is analyzed using Xpower analyzer tool in the Xilinx software. The input supply voltage is Vccint=1.2vVccaux=2.500v and Vcco25=2.5v



Graph 1: Current flow with respect to change in frequency

In Graph 1 the corresponding current flow increases as the clock frequency increases towards the maximum frequency of the system.

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 graph 2 below.



Graph 2: Power consumption with respect to frequency

Performance: To observe the performance of the implementation, a suitable EEG signal (shown in figure 10) 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 Figure 4 and 5. 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.

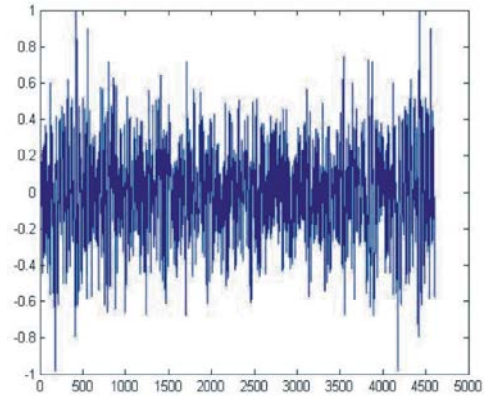


Fig. 3: Input EEG signal

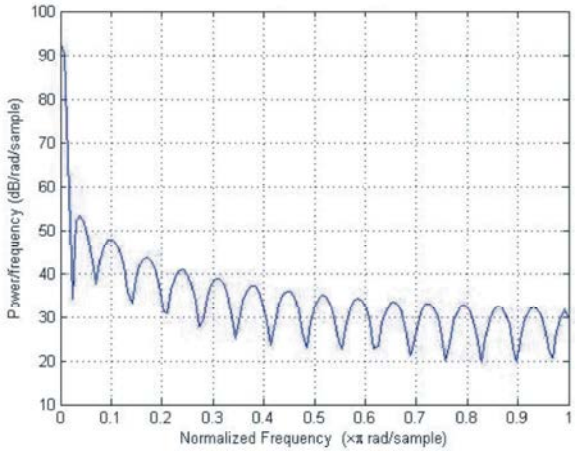


Fig. 4: PSD computed using Matlab.

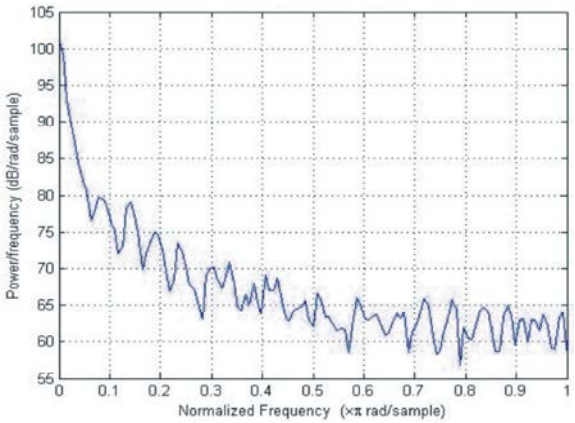


Fig. 5: PSD computed using Xilinx.

Complexity Analysis: The complexity in computation is based on the number of multiplications and number of addition operations required for the complete computation process. In this 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

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 were 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 Mat lab script with the similar function as that of our proposed technique and found an error percentage of 5.87%. The complexity computation confirmed that the technique requires about 55% reduction in multiplication and about 11% reduction in addition when comparing with similar existing works.

REFERENCES

1. Stoica, P. and R.L. Moses, 1997. Introduction to Spectral Analysis.Englewood Cliffs, NJ, USA: Prentice-Hall.
2. Hayes, M., Statistical Digital Signal Processing and Modeling.New York, NY, USA: Wiley.
3. El-Hawary, F. and T. Richards, 1989. A systolic computer architecture for spectrum analysis,"Proc. OCEANS, 4: 1061-1065.
4. Yu, T.H., C.H. Yang, D. Cabric and D. Markovic, 2011. A 7.4 mW 200 MS/s wideband spectrum sensing digital baseband processor for cognitive radios, inProc. Symp. VLSI Circuits, pp: 254-255.
5. Yucek, T. and H. Arslan, 2009. A surveyof spectrum sensing algorithms for cognitive radio applications, IEEE Commun. Surveys Tutorials, 11(1): 116-130.
6. Park, Y., L. Luo, K.K. Parhi and T. Netoff, 2011. Seizure prediction with spectral power of EEG using cost-sensitive support vector machines," Epilepsia, 52(10): 1761-1770.
7. Oppenheim, A. and R. Schafer, XXXX. Discrete Time Signal Processing, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall.
8. Haykin, S., XXXX. Adaptive Filter Theory, 4th ed. Englewood Cliffs, NJ, USA: Prentice-Hall.
9. Welch, P.D., 1967. 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, AU-15: 70-73.
10. Zhang, S., D. Yu and S. Sheng, 2006. A discrete STFT processor for realtime spectrum analysis, inProc. IEEE Asia Pacific Conf. Circuits Syst., pp: 1943-1946.
11. Sanchez, A., M. Garrido, L. Vallejo, J. Grajal and C. Lopez-Barrio, 2005. Digital channelised receivers on FPGAs platforms, inProc. IEEE Int. Radar Conf., pp: 816-821.
12. Garrido, M., K.K. Parhi and J. Grajal, 2009. A pipelined FFT architecture for real-valued signals, IEEE Trans. Circuits Syst. I, Reg. Papers, 56(12) 2634-2643.
13. Ayinala, M. and K.K. Parhi, 2013. FFT architectures for real-valued signals based on radix-23 and radix-24 algorithms, IEEE Trans. Circuits Syst.I, Reg. Papers, pp: 60.
14. Muhammad Abbas, Oscar Gustafsson and and Håkan Johansson, 2013. On the Fixed-Point Implementation of Fractional-Delay Filters Based on the Farrow Structure, IEEE Transactions on Circuits and Systems—I: Regular Papers, 60(4): 926-937.
15. Shuaibing Wu, Ming Wu, Chenxi Huang and Jun Yang, 2012. FPGA-based implementation of steerable parametric loudspeaker using fractional delay filter, Applied Acoustics, 73: 1271-1281.
16. Basant K. Mohanty and Pramod Kumar Meher, 2013. 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).

17. Yu-Chi Tsao and Ken Choi, 2012. 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).
18. Keshab K. Parhi and Manohar Ayinala, 2014. Low-Complexity Welch Power Spectral Density Computation, IEEE Transactions On Circuits And Systems—I: Regular Papers, 61(1).