IETE Journal of Education

ISSN: (Print) 0974-7338 (Online) Journal homepage: http://www.tandfonline.com/loi/tije20

Design and Development of FPGA-Based Spectrum Analyzer

Rupali Borade, Akash Dimber, Damayanti Gharpure & Subramaniam Ananthakrishnan

To cite this article: Rupali Borade, Akash Dimber, Damayanti Gharpure & Subramaniam Ananthakrishnan (2018): Design and Development of FPGA-Based Spectrum Analyzer, IETE Journal of Education, DOI: 10.1080/09747338.2018.1450648

To link to this article: https://doi.org/10.1080/09747338.2018.1450648

Published online: 11 Jun 2018.

IETE JOURNAL OF EDUCATION
https://doi.org/10.1080/09747338.2018.1450648

Design and Development of FPGA-Based Spectrum Analyzer

Rupali Borade, Akash Dimber, Damayanti Gharpure and Subramaniam Ananthakrishnan

Department of Electronic Science, Savitribai Phule Pune University, Pune, India

ABSTRACT

Spectral analysis is important in many signal processing applications as it provides information about the strength of different frequency components present in a time-domain signal. A spectrum analyzer is a tool commonly used to visualize real-world signals in the frequency domain and is capable of performing a variety of signal measurements like power, distortion, harmonics, and noise. This paper discusses design, implementation, and validation of digital spectrum analyzer on the FPGA platform.

Spectral analysis of acquired input data is performed by Fast Fourier Transform (FFT). To enhance spectral analysis accuracy and sensitivity, multiple FFT magnitude outputs are accumulated over windowed sections of the input. Design implementation has been carried out on a Virtex5 xc5vlx50 FPGA device.

KEYWORDS

Data acquisition; FFT; FPGA; Spectrum analyzer; Windowing; Pipeline

1. INTRODUCTION

Spectral analysis plays an important role in many applications, such as communications, radar, oceanography, geophysics, astronomy, atmospheric science, remote sensing, and image processing [1]. While time-domain analysis shows how a signal changes over time, frequency-domain analysis shows how the signal's energy is distributed over a range of frequencies. The spectrum analyzer generates the signal spectrum to provide its frequency-domain features. In recent years, the availability of high-speed, high dynamic range Analog to Digital Converters (ADC) coupled with advancement in Digital Signal Processing (DSP) has changed spectrum analyzer design from conventional analog-based design to digital implementation [2,3].

The algorithm called Fast Fourier Transform (FFT), proposed by Cooley and Tukey [4], made real-time spectrum analysis a practical tool [5]. A Fast Fourier transform (FFT) analyzer calculates the frequency spectrum from a signal captured in the time domain which is digitally sampled by ADC. In regard to the hardware implementation, FPGAs present a powerful tool for development of the digital spectrum analyzer, due to their capability to provide design flexibility, reconfigurability, parallel processing, low power consumption, and high density [6]. The FFT-based spectral analyzer on the FPGA platform has become the method of choice for many implementations [7–12].

An application of reconfigurable computing for the detection of interference using power spectrum analysis has been presented with XCV800 FPGA available on a reconfigurable computing card [7]. Design and implementation of an FFT spectrum analyzer for education purpose have been carried out on a Virtex-II FPGA device (XC2V6000-FF1152) on the Xtreme DSP kit using the system generator software tool [8]. A spectrum analyzer with 1024 FFT points and 20 kHz bandwidth has been realized on Cyclone II EP2C35 FPGA on the Altera DE2 development board [9]. A spectrum analyzer design for 1024-point FFT and 30 MHz bandwidth, together with four selectable windows (Rectangular, Hamming, Blackman-Harris and Flattop), has been implemented on a Virtex-II Xilinx device in the Xtreme DSP kit [10]. An embedded solution for real-time spectrum estimation and analysis based on a modified periodogram of the input signal with 4096 FFT points and 100 MHz bandwidth on Virtex-5 XV5VSX95 T FPGA board has been discussed [11]. Hardware software co-design of the embedded digital spectrum analyzer with 1024 point FFT and 5 MHz bandwidth on the DE0 FPGA board equipped with an Altera Cyclone III 3C16 FPGA device and a VGA display for output spectra is reported [12].

In addition, there has been significant progress over the last decade in the development of FPGA technology that is resistant to and tolerant of the effects of radiation. The success of these efforts has facilitated the use of FPGAs in many existing spacecraft systems [13,14].

This paper presents the design and development of FFT-based spectral analysis on the FPGA platform. The main constraint for designing the FPGA-based spectrum analyzer is its use in the laboratory prototype model of proposed space payload for low-frequency radio astronomical observation below $<20\,\text{MHz}$ [15]. ADC sampling rate is selected in accordance to cover the radio frequency (RF) range of interest. The input signal is sampled with sufficient resolution to preserve its fidelity. Different configurations have been implemented based on requirement of the spectral resolution and input bandwidth by varying the sampling frequency and number of points to be Fourier transformed. It applies the windowing technique to reduce spectral leakage as well as performs frequency binning where multiple FFT's are accumulated to increase the FFT sensitivity. A powerful LABVIEW VI provides a GUI which allows the spectrum to be displayed as well as stored in an excel file in linear or logarithmic scale. The rest of the paper is organized as follows. Section 2 gives the detail about the FFT algorithm. Section 3 gives an overview of the design and implementation of the FPGA-based spectrum analyzer. Section 4 deals with the design optimization. Design testing and calibration are discussed in section 5. The paper ends with the conclusion and future scope in Section 6.

2. FFT ALGORITHM

Discrete Fourier Transform (DFT) used to determine frequency contents of discrete signal sequence $x(n)$ is given by

$$X(k) = \sum_{n=0}^{N-1} x(n)\mathrm{e}^{-\mathrm{j}2\Pi kn/N}, \tag{1}$$

$(k = 0, 1 \ldots .N - 1)$

N-point DFT computation requires N^2 complex multiplications and $N(N{-}1)$ complex additions.

Cooley and Tukey proposed a computationally efficient algorithm called as Fast Fourier transforms which exploits the symmetry and periodicity properties of $\mathrm{e}^{-\mathrm{j}2\Pi/N}$ as shown in Equations (2) and (3) to effectively reduce the DFT computation time [16]

$$\text{Symmetry property:} \quad W_N^{k+N/2} = -W_N^k, \tag{2}$$

where $W_N = e^{-j2\Pi/N}$

$$\text{Periodicity property :} \quad W_N^{k+N} = W_N^k. \tag{3}$$

There are basically two types of FFT algorithms, decimation-in-time FFT algorithms and decimation in frequency FFT algorithms. Decimation-in-time FFT algorithms are based on dividing the input sequence $x(n)$ into smaller and smaller subsequences. Decimation-in-frequency FFT algorithms are based on dividing the output sequence $X(k)$ into smaller and smaller subsequences [17].

2.1. Decimation-in-Time FFT Algorithm

In the case of decimation-in-time FFT algorithm, N-point input sequence $x(n)$ is divided into two $N/2$-point sequences $x_1(r)$ and $x_2(r)$, consisting of even and odd samples of $x(n)$, respectively,

$$x_1(r) = x(2r) \quad \text{and} \quad x_2(r) = x(2r+1), \tag{4}$$

$$\left(r = 0, 1, \ldots \frac{N}{2} - 1\right).$$

N-point DFT can be expressed in terms of the DFTs of the decimated input sequence as follows:

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{nk} \\ &= \sum_{n=0(\text{even})}^{N-1} x(n) W_N^{nk} + \sum_{n=0(\text{odd})}^{N-1} x(n) W_N^{nk} \\ &= \sum_{r=0}^{\frac{N}{2}-1} x(2r) W_N^{2rk} + \sum_{r=0}^{\frac{N}{2}-1} x(2r+1) W_N^{(2r+1)k} \\ &= \sum_{r=0}^{\frac{N}{2}-1} x_1(r) W_{N/2}^{rk} + W_N^k \sum_{r=0}^{\frac{N}{2}-1} x_2(r) W_{N/2}^{rk} \end{aligned} \tag{5}$$

$$X(k) = X_1(k) + W_N^k X_2(k) \tag{6}$$

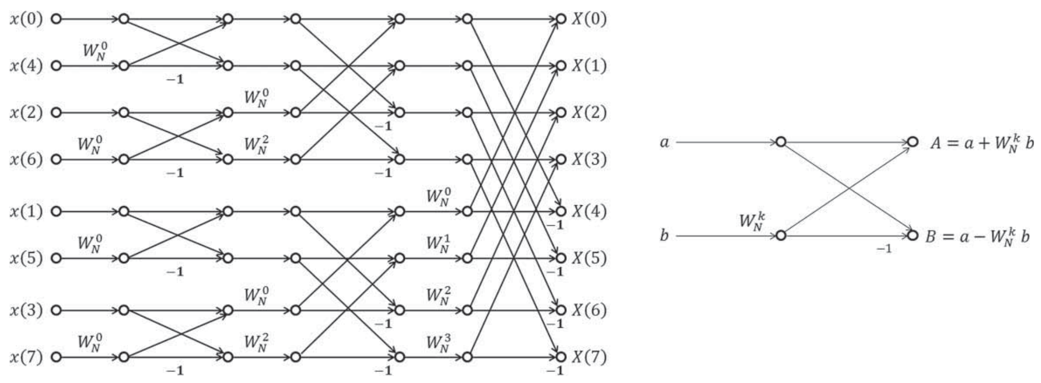
where $X_1(k)$ and $X_2(k)$ are the $N/2$ point DFT of the sequence $x_1(r)$ and $x_2(r)$.

Therefore, $X(k + N/2) = X_1(k + N/2) + W_N^{k+N/2} X_2$ $(k + N/2)$

$$X\left(k + \frac{N}{2}\right) = X_1(k) - W_N^k X_2(k), \tag{7}$$

$$\left(k = 0, 1, \ldots \frac{N}{2} - 1\right)$$

The decimation of data sequence is repeated until the resulting sequences are reduced to a two-point



(a)

(b)

Figure 1: (a) Flow graph of decimation-in-time algorithm for $N = 8$ and (b) basic butterfly computation in the decimation-in-time FFT algorithm

sequence as illustrated for computation of 8-point DFT in Figure 1(a). Basic computation performed at each stage is as shown in Figure 1(b) which is called as butterfly. Each butterfly requires one complex multiplication and two complex additions.

2.2. Decimation-in-Frequency FFT Algorithm

The decimation-in-frequency FFT algorithm is based on calculating the odd and even frequency samples separately. To derive the algorithm, the DFT formula is separated into two summations, one of which contains the sum over the first $N/2$ data points and second sum contains the last $N/2$ data points

$$\begin{aligned}
X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{nk} \\
&= \sum_{n=0}^{(N/2)-1} x(n) W_N^{nk} + \sum_{n=N/2}^{N-1} x(n) W_N^{nk} \qquad (8) \\
&= \sum_{n=0}^{(N/2)-1} x(n) W_N^{nk} \\
&\quad + \sum_{n=0}^{(N/2)-1} x\left(n + \frac{N}{2}\right) W_N^{(n+N/2)k} \\
&= \sum_{n=0}^{(N/2)-1} \left[x(n) + W_N^{(N/2)k} x\left(n + \frac{N}{2}\right) \right] W_N^{nk} \\
&= \sum_{n=0}^{(N/2)-1} \left[x(n) + (-1)^k x\left(n + \frac{N}{2}\right) \right] W_N^{nk}, \\
&(k = 0, 1 \dots .N - 1)
\end{aligned}$$

$X(k)$ can be split into the even- and odd-numbered samples

$$X(2r) = \sum_{n=0}^{(N/2)-1} \left[x(n) + x\left(n + \frac{N}{2}\right) \right] W_{N/2}^{nr}, \qquad (9)$$

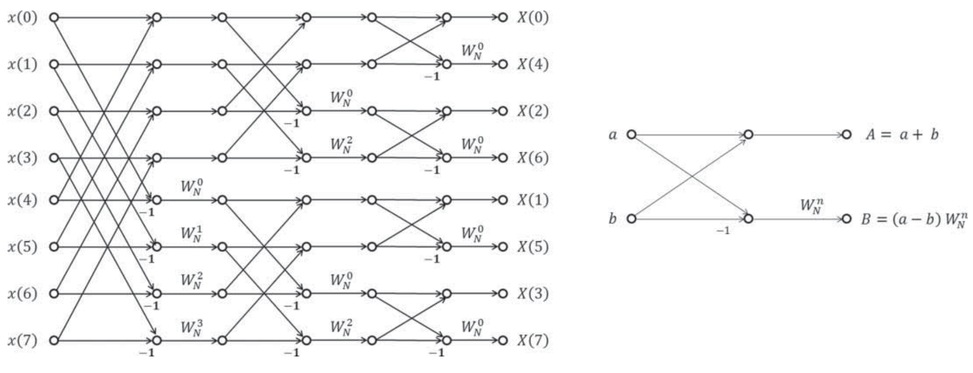
$$\left(r = 0, 1, \dots \frac{N}{2} - 1 \right),$$

$$X(2r + 1) = \sum_{n=0}^{(N/2)-1} \left[x(n) - x\left(n + \frac{N}{2}\right) \right] W_N^{n} W_{N/2}^{nr}, \qquad (10)$$

$$\left(r = 0, 1, \dots \frac{N}{2} - 1 \right).$$

This computation is repeated through decimation of the $N/2$ point DFTs $X(2r)$ and $X(2r + 1)$. Illustration of 8-point decimation in the frequency algorithm is shown in Figure 2(a). The butterfly computation in decimation in the frequency FFT algorithm is shown in Figure 2(b). Each butterfly requires one complex multiplication and two complex additions.

The Cooley–Tukey algorithm considers FFT size N to be powers of base r, such that $N = r^v$ and the algorithms are termed as Radix-r FFT algorithms. In a Radix-r FFT algorithm, the butterfly block performs r-point FFT. Using butterflies of higher radices will reduce the number of stages and butterflies, but will increase the complexity of butterfly. Most common radices are 2, 4 and 8. Radix-2 algorithms are widely used due to the simplicity of the butterfly [18].



(a) (b)

Figure 2: (a) Flow graph of decimation-in-frequency algorithm for $N = 8$ and (b) basic butterfly computation in the decimation-in-frequency FFT algorithm

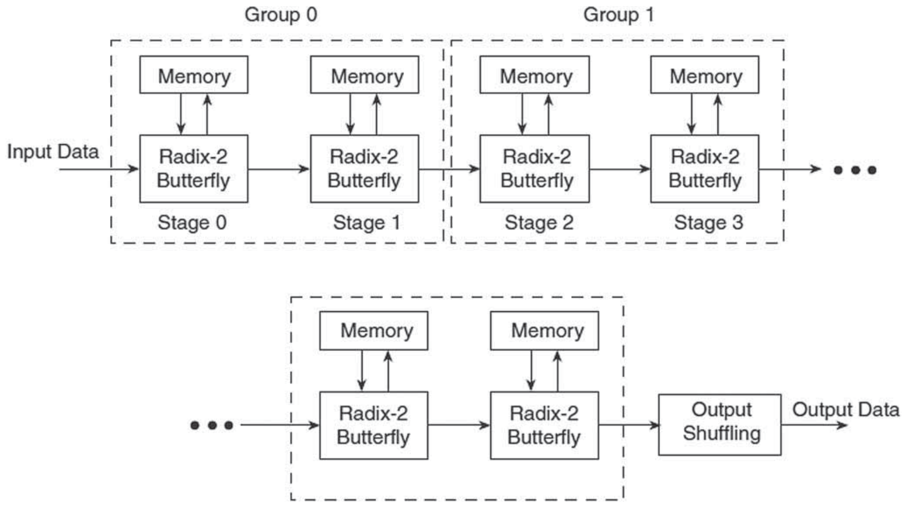


Figure 3: Pipelined streaming I/O [19]

The 8 point FFT described in Figures 1 and 2 is an example of the Radix 2 FFT algorithm. The number of complex multiplications and complex additions required to compute N-point DFT using the radix-2 FFT algorithm are $(N/2)\ \log_2 N$ and $N\ \log_2 N$ respectively.

2.3. FFT Algorithm Implementation

In the design, the FFT IP core from Xilinx is used which implements the Cooley–Tukey FFT algorithm. Xilinx FFT IP provides four architectures such as Pipelined Streaming I/O, Radix-4 Burst I/O, Radix-2 Burst I/O, and Radix-2 Lite Burst I/O. These architectures offer a trade-off between core size and transform time. Out of these architectures, Pipelined, Streaming I/O architecture based on decimation in frequency, provides better throughput at the cost of higher FPGA resources and is selected to meet real-time constraints. The Pipelined Streaming I/O solution, pipelines several Radix-2 butterfly processing engines to offer continuous data processing as shown in Figure 3 [19]. Each processing engine has its own memory banks to store the input and intermediate data.

3. DESIGN AND IMPLEMENTATION

Design of the spectrum analyzer has been realized on a VIRTEX-5 protoboard (MX5VFK-LX50). The analog signal is sampled and digitized withLTC 2255; 14 bit, 100 MSPS ADC add on card, available on board as shown in Figure 4. Other peripherals used in the design are RS232 serial interface to communicate with host PC, switches for testing and LEDs for indication.

Various modules required are implemented in VHDL language using the Xilinx ISE Design Suite. Acquired input data go through different processing stages as shown in Figure 5.

Before performing FFT, input samples are preprocessed by multiplying them with the window function to reduce spectral leakage in FFT estimation of the finite-length time sequence. Window function amplitude decreases

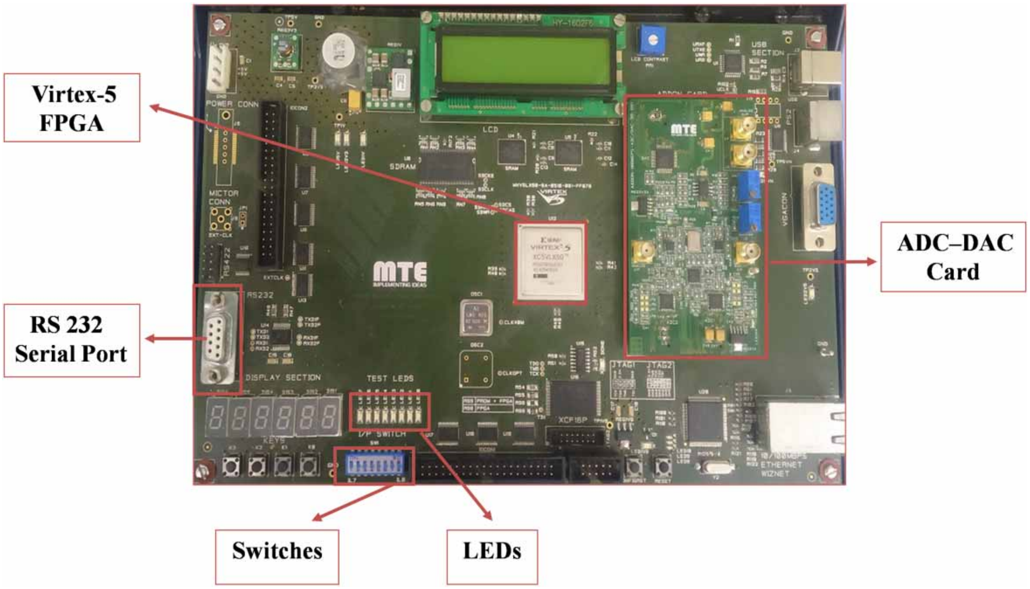


Figure 4: Virtex — 5 development board

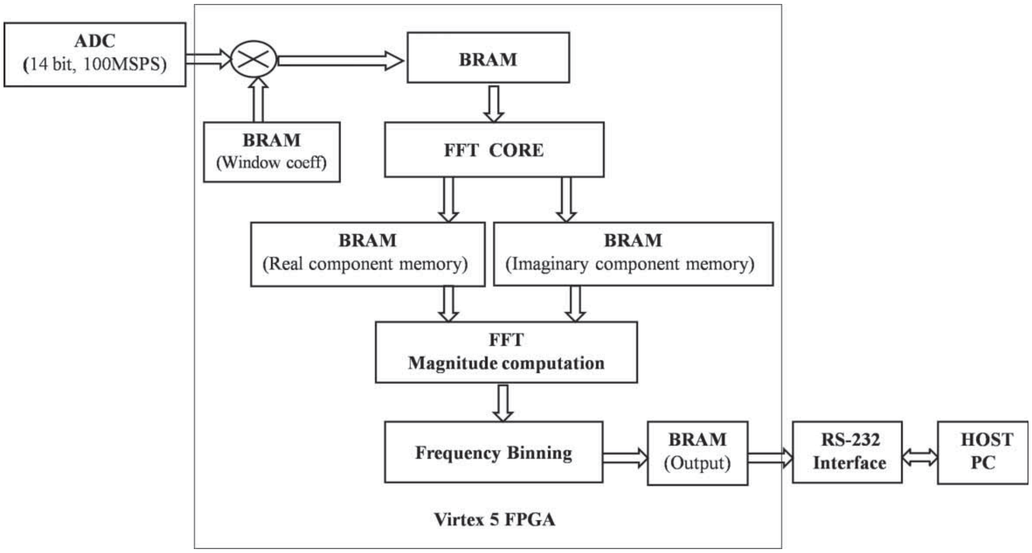


Figure 5: Data flow diagram of FPGA-based spectrum analyzer

smoothly towards zero value at the boundaries. This reduces or eliminates the discontinuities at the boundaries reducing the spectral leakage [20,21]. Figure 6 shows the spectrum of a typical window function consisting of a main lobe and various side lobes.

The side lobe behaviour of the window function affects the extent to which adjacent frequency components leak into adjacent frequency bins. The ability to detect weak components in a spectrum is affected by the highest side lobe level and the side lobe fall-off rate and the frequency resolution is limited by the main lobe width of the window. Therefore, the ability to distinguish two closely spaced frequency components of equal amplitude increases as the main lobe of the window narrows [22].

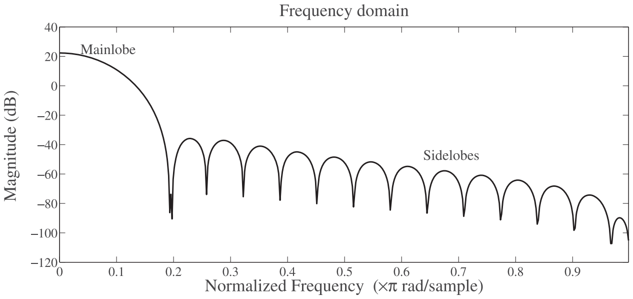


Figure 6: Spectrum of a typical window function

There is fundamental trade-off between the main lobe width and the side lobe level. Decreasing the side lobes for reduced spectral leakage increases the main lobe width causing lower frequency resolution [20–25]. For radio astronomical application, the Hanning window is chosen due to its good frequency resolution and faster fall-off of the side lobes [26,27].

Window coefficients generated using MATLAB are stored in Xilinx BRAM instantiated as ROM. The coefficients are scaled appropriately to fit into unsigned integer word type to 11 bits. With the 14 bits input samples $x(n)$ and 11-bits window coefficients $w(n)$,windowing results in 25 bit output, which is truncated and stored as 14 bits in BRAM.

Windowed input data are fed to the FFT core which is the central element of the system, performing the conversion from time-domain to the frequency-domain. Output data width of the FFT core with an input of 14 bits, 1024-point FFT transform has real and imaginary output of 25 bits which are passed to the magnitude block. Mathematical operations required for FFT magnitude computations are performed using the Xilinx in-built multiplier, adder, and square root IP core. The FFT magnitude is truncated to 24 bits due to the limited input size of the square root generator. The FFT output is in the form of discrete frequencies, *i.e.* bins that represent collective magnitude or energy from a small range of frequencies based on resolution (f_s/N).

Low-frequency radio astronomical signals are weak signals with very poor signal-to-noise ratio. The final step of processing consists of frequency binning; where magnitudes of multiple FFTs are accumulated in 32-bit output memory to reduce the random noise fluctuations and to improve detection of weak signals. Frequency binning is the averaging the magnitudes of multiple FFTs. In averaging, each instantaneous spectrum is added to the next (bin wise) and the sum is divided by the total number of spectra added, *i.e.* to average k number of N-point FFTs, For example, any of the ith bin of frequency binned FFT $\times$ (i) is given by

$$X(i) = \frac{|X_1(i)| + |X_2(i)| + |X_3(i)| + \cdots + |X_k(i)|}{k},$$

where $X_n(i)$ is the magnitude of the ith bin from the the nth FFT. Frequency binning helps to reduce the random noise fluctuations in the FFT spectrum and improves detection of weak signals [25]. Division by k of the accumulated FFT magnitude spectrum is performed on the PC with the help of LABVIEW VI developed.

The single channel FPGA-based spectrum analyzer utilizes about 40% of Virtex-5 FPGA resources as detailed in Table 1.

Table 1: Device utilization summary

Slice logic utilization	Used	Available	Utilization (%)
Number of slice registers	4712	28,800	16
Number of slice LUTs	6285	28,800	21
Number of occupied slices	2195	7,200	30
Number of bonded IOBs	28	440	6
Number of DSP48Es	17	48	35

Table 2: Processing time of different stages for 1024 point FFT with 100 MHz sampling frequency

Design stages	Processing time (μs)
Data acquisition and windowing	10
FFT computation	20
FFT magnitude calculation	20
FFT magnitude accumulation in one cycle	5

The UART-RS232 interface is implemented to test the system as well as to transfer the data to the LabVIEW GUI. For debugging purposes, the outputs of each processing stage, namely windowed input, FFT real and imaginary parts, FFT magnitude and accumulated FFT output, are stored in a dual port BRAM, which can be accessed through the RS232 port. This simple design for testability introduced makes it possible to obtain debugging information at any stage of the processing.

4. DESIGN OPTIMIZATION

Once the basic modules were simulated and tested, the next task was to improve the performance spectrum analyzer in terms of processing speed and spectral resolution.
4.1. Speed Optimization

Design optimization for speed can be achieved by reducing the processing time of different modules. Xilinx FFT IP provides four architecture options to offer a trade-off between core size and transform time. Various architectures available were implemented and their features were studied. Out of these architectures, the Pipelined, Streaming I/O architecture is selected to meet real-time constraints, *i.e.* latency and FPGA resources. The IP cores used for magnitude computation are also optimized for speed. In addition, exploiting the symmetry property of Fourier transform, computation of FFT magnitude calculation, frequency binning, etc. is carried out with N/2, *i.e.* 512 points, which reduces storage and computation time. Table 2 summarizes timing of each stage for acquisition and processing of 1024 point input data with 100 MHz sampling frequency. Processing time is estimated by observing the number of clock cycles utilized by each processing stage on simulation waveform generated using the Xilinx ISim simulator.

Initially, processing and accumulation of multiple FFTs was carried out sequentially as in Figure 7, where, next data acquisition cycle starts after complete processing on previously acquired data. That resulted in a time interval of 45 μs between the consecutive data acquisition periods of 10 μs.

The concept of pipelining has been implemented to improve the speed of processing and to reduce the interval between two data acquisition periods. The data flow with pipelining is as shown in Figure 8, wherein next data acquisition starts just after completion of the data loading phase of the FFT core.

It can be observed that pipelining of stages reduces the time gap between data acquisition stage from 45 to 10 μs and provides an FFT output after every 20 μs. That results in a computation of 50,000, 1024-point FFTs/s against the earlier rate of 20,000 FFTs/s.

4.1. Improvement in Spectral Resolution

The FPGA-based spectrum analyzer was designed as a prototype model of the proposed space payload, for low-frequency radio astronomical observation in the range of 0.2–16 MHz. The front end consists of antenna, followed by a low-noise amplifier and a band pass filter to cover the RF range of interest. Bandwidth adaptation is taken care of by the bandpass filter designed in the RF front end.

Table 3: FPGA-based spectrum analyzer performance for different FFT size and sampling frequency

Case	Sampling frequency (MHz)	Spectral resolution (kHz)	Bandwidth (MHz)	FFTs/s
(I) $N = 1024$	100	97.65	50	50,000
	50	48.82	25	33,333
	33.33	32.54	16.67	25,000
(II) $N = 2048$	100	48.82	50	25,000
	50	24.41	25	16,667
	33.33	16.27	16.67	12,500

The spectrum analyzer designed for 1024 point FFT with 100 MHz sampling frequency provides spectral resolution of (f_s/N) 100 MHz /1024 = 97.65 kHz. For further improvement in spectral resolution, FFT size is increased to 2048 point and the sampling rate is reduced up to 33.33 MHz by down sampling. Table 3 outlines the performance of different design configurations measured in terms of number of FFTs/s, spectral resolution, available bandwidth, etc. Increased FFT size with reduced sampling rate increases the input acquisition time and FFT processing time. Therefore, the rate of number of FFT computed per second decreases from 50,000 to 12,500 per second.

5. TESTING AND CALIBRATION

At first, functionality of each module used in the design such as ADC-memory interface, windowing, FFT calculation was tested independently and then these modules were integrated to realize the main system. For primary

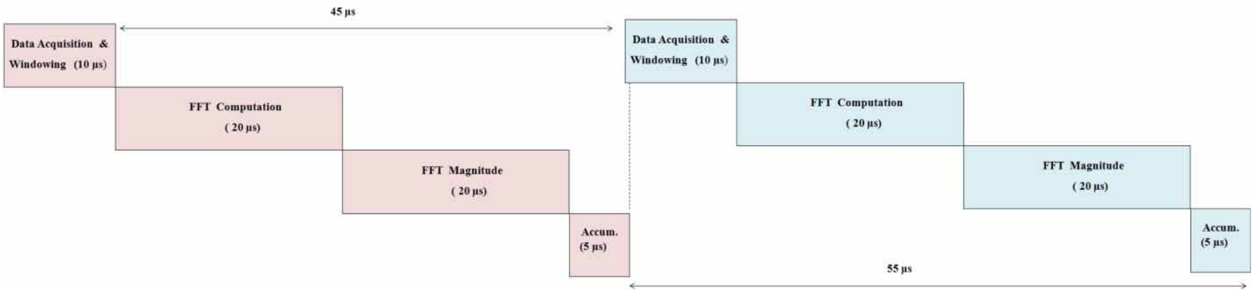


Figure 7: Sequentially execution of processing stages

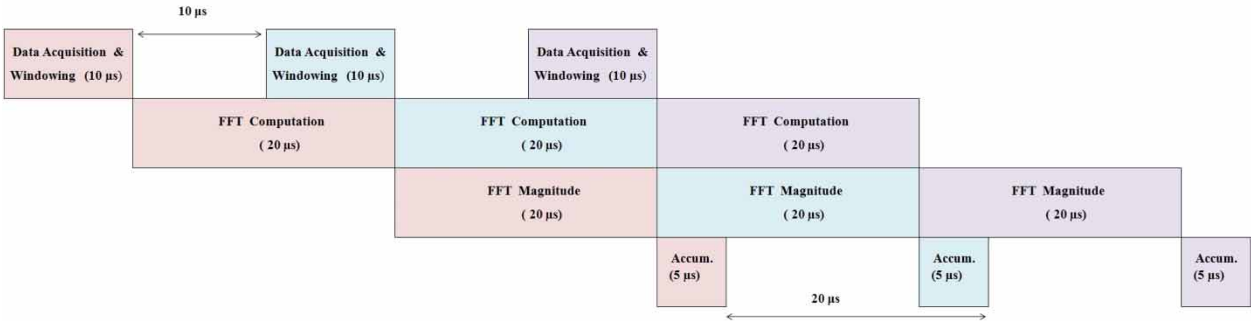


Figure 8: Pipelining of processing stages

testing of the complete design, input memory was initialized with different test signals generated from MATLAB. Data flow and synchronization between different modules were observed and verified on simulation waveform. For step wise debugging of the implemented design, each processing stage output stored in the BRAM has been read through the RS232 interface. LABVIEW VI has been developed extensively for testing, which allows one to load the data and command to the system for testing purpose. The LABVIEW VI reads data stored in the BRAM's through the RS232 port and displays the same.

To evaluate the FPGA-based spectrum analyzer, based on the sampling rate and input range of ADC, sine wave signals of different frequencies and amplitudes were applied as input to the FPGA spectrum analyzer using the Agilent N9310A RF signal generator as shown in Figure 9. FPGA results are observed on LABVIEW VI which receives FFT output data, converts it into dB scale and displays as well as stores it into the excel file. The spectrum obtained with the FPGA-based spectrum analyzer is also verified by comparing it with the Agilent N1996A spectrum analyzer.

At frequencies which are not integer multiple of spectral resolution; spectral leakage was observed where signal amplitude leaks from one frequency bin to another. To illustrate spectral leakage, FFT results for 10 MHz and for 12.5 MHz frequency obtained in the design of 2048 point FFT with 50 MHz sampling frequency are shown in Figure 10 (a) and 10(b) respectively. It is observed that spectral leakage occurs at 10 MHz frequency, whereas there is a sharp peak at 12.5 MHz.

The effect of windowing is shown in Figure 11, where the 10 MHz sine wave signal spectrum for the rectangular (*i.e.* without window) and for Hanning window are displayed and the improvement is clearly seen.

To calibrate the spectrum analyzer, amplitude of a sine wave signal of fixed frequency was varied over the range of −80 to + 10 dBm based on the ADC input range. In single FFT, due to random noise fluctuation signal level

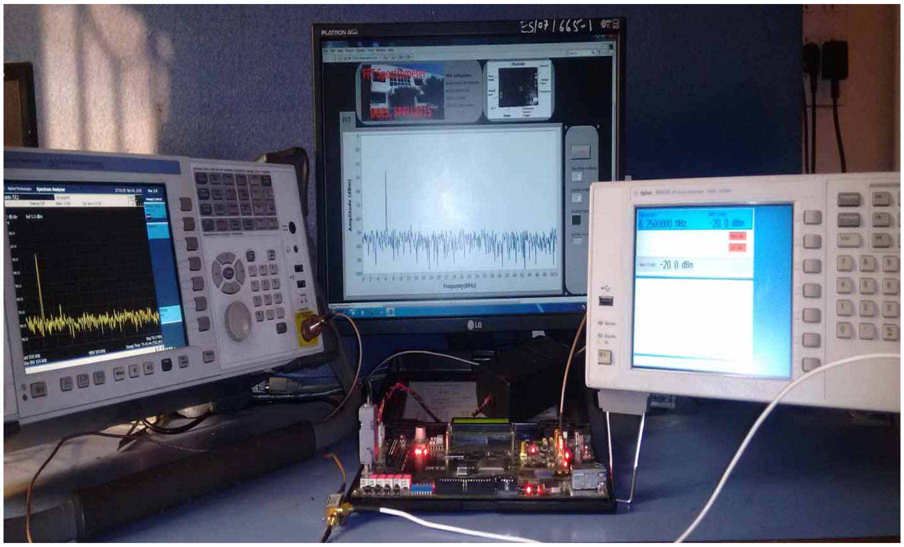
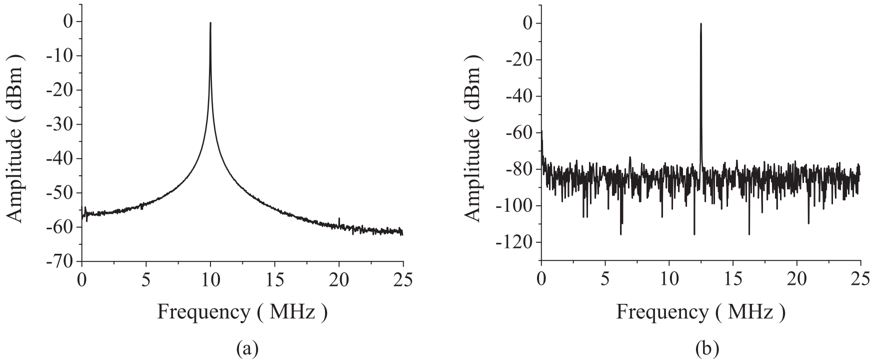


Figure 9: FPGA-based spectrum analyzer test set-up



(a)

(b)

Figure 10: FFT result: (a) for 10 MHz sine wave signal and (b) for 12.5 MHz sine wave signal

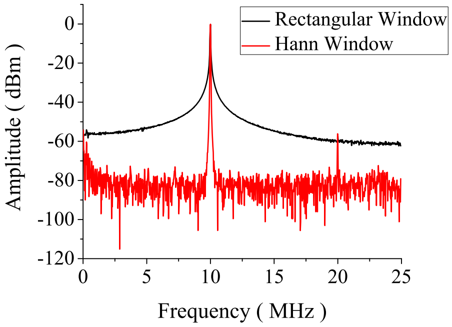


Figure 11: 10 MHz sine wave signal FFT output with a rectangular and Hanning window

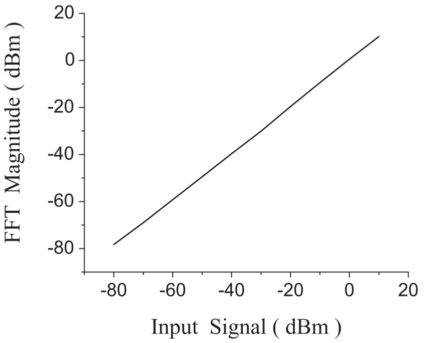


Figure 13: Calibration of FPGA-based spectrum analyzer

below −70 dBm could not be clearly detected as a peak at particular frequency. In order to increase FFT sensitivity, fluctuations in noise is reduced by accumulation of multiple FFT magnitudes. Multiple FFT accumulation helps to identify signal level up to −80 dBm. As shown in Figure 12(a) for single FFT computation, −80 dBm sine wave signals could not be detected as it was buried in the noise. The same signal peak is clearly visible in Figure 12(b) after accumulation of 8192 FFTs. Maximum number of spectra to be accumulated depends on the size of output memory. For this case, use of 32-bit memory allows accumulation of 8192 spectra.

After testing and calibration, it was observed that the FPGA-based spectrum analyzer could digitize and process signals with levels in the range −80 to +10 dBm over the defined bandwidth. It was observed that FFT magnitudes vary linearly with respect to the input signal levels as shown in Figure 13.

5.1. Testing the FPGA-Based Spectrum Analyzer with Active Antenna

A number of experiments were carried out to test the spectrum analyzer implemented, by applying standard test signals like sine wave, square wave generated using the Agilent signal generator varying frequency over the band of interest. For calibration purposes, the test signal with power over the range of +10 to −80 dBm were applied as input and an absolute error of 0.5 dBm and relative error of about 1.25% were observed in the FPGA spectrum analyzer output.

A 1-meter long monopole active antenna, designed for low-frequency radio astronomy observation was used for testing as shown in Figure 14. Signals picked up by the antenna are filtered out with band pass filter of 0.2–20 MHz to select the desired range of frequencies. To validate the output of the FPGA-based spectrum analyzer, the results are compared with the Agilent N1996A spectrum analyzer. For this purpose, the output of the filter was split using a Mini Circuits ZX10R-14-S+ splitter, and one output is fed to the Agilent spectrum analyzer and the other to the FPGA-based spectrum analyzer.

For the purposes of these measurements, the resolution bandwidth of the Agilent spectrum analyzer was kept at 25 kHz; the frequency span at 25 MHz with 1024 sweep points to match the specification of the FPGA-based spectrum analyzer. The results from both instruments are

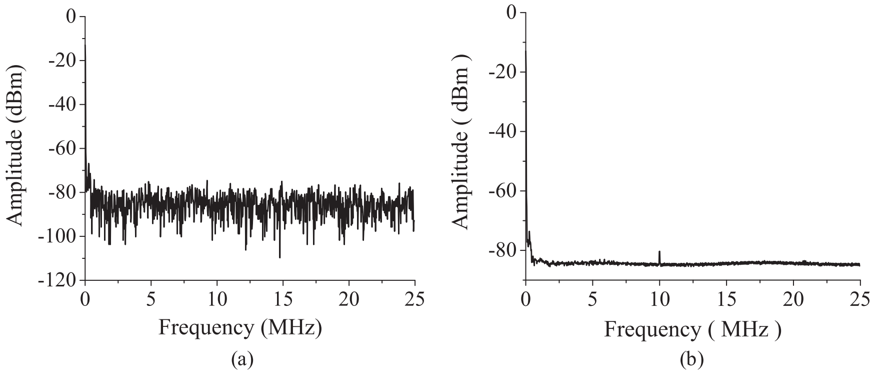


Figure 12: Spectrum of −80dBm sine wave signal at 10 MHz frequency: (a) for single FFT computation and (b) for multiple FFT magnitude accumulation

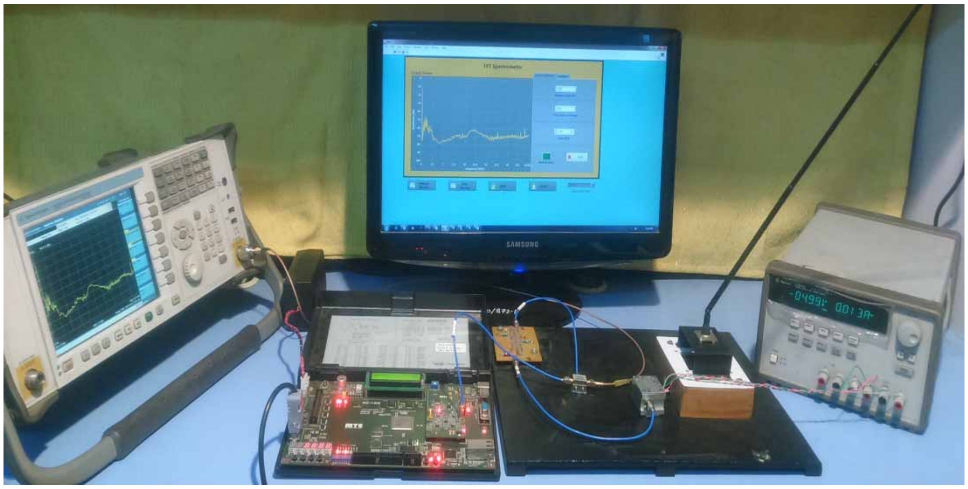


Figure 14: Antenna test set-up

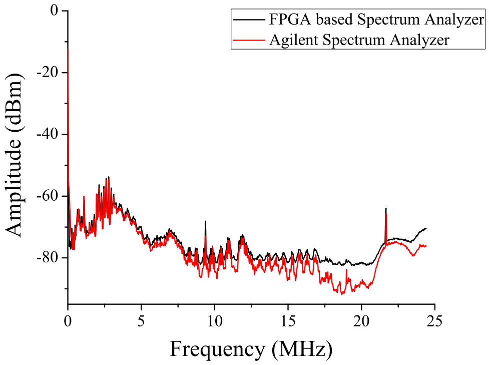


Figure 15: Spectrum of local environmental noise picked up by the monopole antenna without amplifier

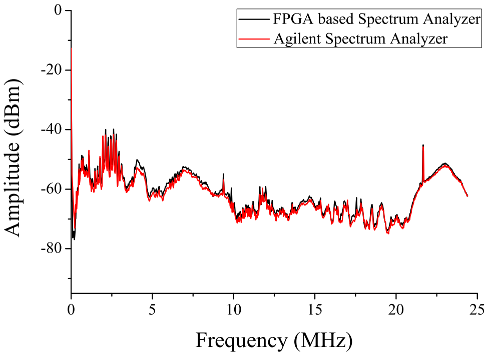


Figure 16: Spectrum of local environmental noise picked up by the monopole antenna with amplifier

shown in Figure 15. It must be noted that the minimum signal level detected by the FPGA backend is −80 dBm, whereas the Agilent spectrum analyzer can detect the signal level up to −100 dBm.

Further, the experiments were repeated by connecting the output of the antenna to the matching amplifier. With the matching amplifier having 20 dB gain, the amplitude of the entire spectrum was observed to be shifted nearly by 20 dB, *i.e.* the gain of the amplifier as shown in Figure 16. The spectrum from the FPGA-based spectrum analyzer and Agilent spectrum analyzer shows excellent agreement with a correlation factor of 0.99.

It must be noted here that the FPGA spectrum analyzer binned 8192 spectra over 491.5 ms, while the Agilent spectrum analyzer averaged 100 passes over 9.2 s which accounts for the differences in the two spectra. Some frequency components are observed in the FPGA spectrum analyzer output which might be averaged out by the Agilent spectrum analyzer due to the longer period of observation, *i.e.* 9.2 s. For frequency components, which are prominent in the signal received by antenna, maximum absolute error of 2 dBm and maximum relative error of 4% are observed between the FPGA and Agilent spectrum analyzer.

5.2. Specifications of the FPGA-Based Spectrum Analyzer

The preliminary requirements for the design are the observation frequency range below 20 MHz with a minimum spectral resolution and dynamic range of 80 dB. Therefore, a 14 bit ADC, 33 MHz sampling rate is selected with the use of windowing technique and frequency binning algorithm to achieve required dynamic range. Table 4 summarizes the specifications of the FPGA-based spectrum analyzer; minimum spectral resolution of 16 kHz is achieved with 2048 point FFT and 33 MHz sampling frequency.

Table 4: Specification of FPGA-based spectrum analyzer

Parameter	Result, unit
Number of input channels	1
Input sampling rate	33 MHz
Input bandwidth	16 MHz
ADC resolution	14 bit
Spectral resolution	16 kHz
Window function	Hann
Input range	−80 to +10 dBm

6. CONCLUSION AND FUTURE SCOPE

In this paper, design and development of FFT-based digital spectrum analyzer for the prototype model of low-frequency radio astronomy observation have been discussed on the FPGA platform. Design has been successfully implemented on the Xilinx Virtex5 based xc5vlx50 FPGA device using Xilinx software and VHDL language.

Spectral resolution of 16 kHz is achieved with 2048 points FFT and 33 MHz sampling frequency. The windowing technique is used to reduce the spectral leakage inherent in FFT estimation. Design is optimized in terms of speed by developing pipeline of processing stages which provides 50,000 FFT/s. The concept of frequency binning has been successfully utilized to increase the sensitivity of the spectrum analyzer to −80 dBm. Also the LabVIEW-based GUI designed provides a menu-driven interactive user interface. The LabVIEW VI allows the user to set the downsampling factor and option for single time or accumulations of multiple FFTs.

The FFT-based spectrum analyzer implemented on the FPGA platform is designed for the proposed space payload to study low-frequency radio astronomical observation. The FPGA-based spectrum analyzer can easily be used for many other real-world applications and for education purpose as well. It was successfully used for antenna characterization and radio noise measurements. Moreover, these measurements were also carried out at remote places by making the FPGA-based spectrum analyzer standalone. It has also been used to test the performance of various electronics devices such as LNA, filters, noise generator, *etc*. It is also applicable for ADC characterization where output samples from an ADC are processed with an FFT to measure the integral nonlinearity, distortion, and signal-to-noise ratio. For education purpose, the FPGA-based spectrum analyzer provides a low-cost platform to understand frequency-domain behaviour of various signals as well as to study different signal processing concepts. Further, efforts are being made for continuous data acquisition and applying the overlapping window technique to minimize the signal loss due to the window edge effect. Implementation of modules to incorporate reconfigurable FFT hardware in terms of FFT size, window function, and number of FFTs to be accumulated is also being carried out.

FUNDING

This work was supported by Indian Space Research Organization (ISRO) India [Grant Number GOI-A-574].

REFERENCES

1. D. B. Percival, and A. T. Walden, *Spectral Analysis for Physical Applications*. Cambridge, UK: Cambridge University Press, 1993.

2. M. T. Hunter, A. G. Kourtellis, C. D. Ziomek, and W. B. Mikhael, “Fundamentals of modern spectral analysis,” *IEEE Instrum. Meas. Mag.*, Vol. 14, no. 4, pp. 12–16, Aug. 2011.

3. S. M. Kay, and S. L. Marple, Jr., “Spectrum analysis – a modern perspective,” *Proc. IEEE*, Vol. 69, no. 11, pp. 1380–419, Nov. 1981.

4. J. W. Cooley, and J. W. Tukey, “An algorithm for the machine calculation of complex Fourier series,” *Math. Comput.*, Vol. 19, pp. 297–301, Apr. 1965.

5. S. C. Dutta Roy, “The ABCD’s of digital signal processing (Part 1),” *IETE J. Educ.*, Vol. 21, no. 1, pp. 3–12, 1980.

6. G. Stitt, “Are field-programmable gate arrays ready for the mainstream?” *IEEE Micro*, Vol. 31, no. 6, pp. 58–63, Nov./Dec. 2011.

7. Y. Abhyankar, C. Sajish, Y. Agarwal, C. R. Subrahmanya, and P. Prasad, “High performance power spectrum analysis using a FPGA based reconfigurable computing platform,” in *Proceedings of the IEEE International Conference on Reconfigurable Computations. FPGA’s (ReConFig)*, San Luis Potosi, Mexico, Sep. 2006, pp. 1–5.

8. T. Sansaloni, A. Perez-Pascual, V. Torres, V. Almenar, J. F. Toledo, and J. Valls, “FFT spectrum analyzer project for teaching digital signal processing with FPGA devices,” *IEEE Trans. Educ.*, Vol. 50, no. 3, pp. 229–35, Aug. 2007.

9. Vigil Bárbaro M. López-Portilla, et al., “FPGA based spectrum analyzer.,” in *Argentine Congress of Embedded Systems*, Buenos Aires, Argentina, Aug. 2013.

10. Digital Communications Laboratory, UniversitatPolitècnica de València. “FPGA-based vectorial spectrum analyzer,” . Available: http://www.gised.upv.es/anaspect.html, accessed 2015.

11. V. Iglesias, J. Grajal, M. A. Sánchez, and M. López-Vallejo, "Implementation of a real-time spectrum analyzer on FPGA platforms," *IEEE Trans. Instrum. Meas.*, Vol. 64, no. 2, pp. 338–55, Feb. 2015.

12. D. Qi, X. Guo, and W. Du, "Design of digital signal spectrum analyzer based on FPGA," in *IEEE International Conference on Information and Automation*, Lijiang, China, Aug. 2015, pp. 2725–9.

13. P. J. Pingree, D. L. Bekker, T. A. Werne, and T. O. Wilson, "The prototype development phase of the CubeSat On-board processing Validation Experiment," *Aerospace Conference, 2011 IEEE*, Big Sky, MT, 2011, pp. 1–8.

14. D. L. Jones, T. J. W. Lazio, and J. O. Burns, "Dark Ages Radio Explorer mission: Probing the cosmic dawn," *2015 IEEE Aerospace Conference*, Big Sky, MT, 2015, pp. 1–8.

15. K. Makhija, R. Borade, G. Shaifullah, S. Gujare, S. Ananthakrishnan, and D. C. Gharpure, "Space electromagnetic and plasma sensor (SEAPS): A laboratory prototype for a space payload," *MAPAN*, Vol. 31, no. 4, pp. 283–9, Dec. 2016.

16. J. G. Proakis, and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1996, pp. 448–75.

17. A. V. Oppenheim, and R. W. Schafer, *Discrete-Time Signal Processing*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999, pp. 629–50.

18. M. L. Ferreira, A. Barahimi, and J. C. Ferreira, "Reconfigurable fpga-based FFT processor for cognitive radio applications," in *International Symposium on* Applied Reconfigurable Computing, Springer, March 2016, pp. 223–32.

19. Xilinx, Available: http://www.xilinx.com/support/documentation/ip_doumentation/ds808_xfft.pdf

20. F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proc. IEEE*, Vol. 66, no. 1, pp. 51–83, Jan. 1978.

21. A. H. Nuttall, "Some windows with very good sidelobe behavior," *IEEE Trans. Acoust., Speech, Signal Process.*, Vol. 29, pp. 84–91, Feb. 1981.

22. S. Rapuano and F. J. Harris, "An introduction to FFT and time domain windows," *IEEE Instrum. Meas. Mag.*, Vol. 10, no. 6, pp. 32–44, Dec. 2007.

23. M. Cerna, and A.F. Harvey, "The fundamentals of FFT-based signal analysis and measurement", National Instruments Corporation, Application Note 041, 2000.

24. T. H. Yoon and E. K. Joo, "A flexible window function for spectral analysis [DSP tips & tricks]," *IEEE Signal Process. Mag.*, Vol. 27, no. 2, pp. 139–42, 2010.

25. R. G. Lyons, *Understanding Digital Signal Processing*. Upper Saddle River, NJ: Pearson Education, 2010, pp. 135–60, 671–730.

26. A. W. Hotan, "A new signal processing platform for radio astronomy," *Astron. Astrophys.*, Vol. 485, pp. 615–622, July 2008.

27. V. B. Ryabov, D. M. Vavriv, P. Zarka, B. P. Ryabov, R. Kozhin, V. V. Vinogradov, and L. Denis, "A low-noise, high-dynamic-range, digital receiver for radio astronomy applications: an efficient solution for observing radio-bursts from Jupiter, the Sun, pulsars, and other astrophysical plasmas below 30 MHz," *Astron. Astrophys.*, Vol. 510, p. A16, Feb. 2010.

Authors

Rupali Borade received the B.E. degree in Electronics Engineering from A.I.S.S.M.S College of Engineering, Pune University, India, in 2010, and M.Tech degree in Electronics from Walchand College of Engineering, Sangli, India, in 2013. She is currently working as Senior Research Fellow at Savitribai Phule Pune University. Her research interests are in the field of DSP and FPGA-based design.

Corresponding author. Email: rup.1889@gmail.com.

Akash Dimber received the B.Sc. degree in Electronics from Nowrosjee Wadia College, India, in 2013, and M.Sc. degree in Instrumentation from the Pune University, India, in 2015. He is currently working as Project Assistant at Savitribai Phule Pune University. His research interests are in the field of Embedded Electronics and RF Analog Electronics.

Email: aakash.dimber@gmail.com

Damayanti Gharpure is working as a Professor in the Department of Electronic Science, Savitribai Phule Pune University and has 30 years of experience in research and teaching. She has been working in the field of Intelligent system design starting right from microprocessor-based/PC based systems to microcontroller/FPGA-based embedded systems. Development of necessary software and the design of indigenous hardware to make system more capable/intelligent was the goal pursued. She has worked on a number of research projects and consultancy projects. Her areas of interest are image processing and analysis, pattern recognition, applications of artificial neural network, odour sensing and intelligent system design. She has published a number of papers in national and international journals.

Email: dcg@electronics.unipune.ac.in

Subramaniam Ananthakrishnan is currently an Adjunct Professor of the Savitribai Phule Pune University, and a Raja Ramanna Fellow, formerly a Senior Professor of the Tata Institute of Fundamental Research (TIFR) with nearly 50 years of experience in radio astronomy. He has made significant contribution to the building up of large radio astronomy instruments in India like Ooty Radio Telescope and the GMRT. His research work spans solar wind plasma, comet tail scintillation, solar and extragalactic radio astronomy and Very Long Basline Interferometry in which he has published about 85 papers in Indian and foreign journals. A fellow of all the National Science Academics of India, he is very keen in attracting students to the area of radio astronomy, remote sensing, and space physics, all of which requires a skill set involving physics, mathematics, electronics, and software.

Email: subra.anan@gmail.com