Nigerian Journal of Technology (NIJOTECH)
Vol. 45, No. 1, March 2026, pp. 1- 16
www.nijotech.com

Print ISSN: 0331-8443
Electronic ISSN: 2467-8821
https://doi.org/10.4314/njt.2026.3713

DESIGN OF AN FPGA BASED MULTI-FEATURE SPECTRUM ANALYZER

AUTHORS:
A. Yusuf[1], *P. Y. Dibal[1], C. U. Ngene[1] and J. Okpor[2]

AFFILIATIONS:
[1]Department of Computer Engineering, University of Maiduguri, NIGERIA.
[2]Department of Computer Engineering, Federal University Wukari, Taraba, NIGERIA.

***CORRESPONDING AUTHOR:**
Email: yoksa77@gmail.com

ARTICLE HISTORY:
Received: March 16, 2024.
Revised: July 07, 2025.
Accepted: January 24, 2026.
Published: Early view

KEYWORDS:
Spectrum Analyzer; Very High Speed Integrated Circuit Hardware Description Language (VHDL); Spectrum; Fast Fourier Transform (FFT).

ARTICLE INCLUDES:
Peer review

DATA AVAILABILITY:
On request from author(s)

EDITORS:
Chidozie Charles Nnaji

FUNDING:
None

Abstract

A spectrum analyzer is a device that plays central roles in many signal processing, machine monitoring, and diagnostics applications. It is used in evaluating the performance or functionality of a system through the analysis of the spectrum features of the signals from the system under observation. In this paper, a spectrum analyzer (SA) was designed for implementation on a field programmable gate array (FPGA); the SA was designed to output four important signal spectrum features- frequency, amplitude, phase, and power. The design approach in this work involved the design of an architecture for the SA, and coding implementation and verification using a combination of a high level synthesis (HLS) based on C language, and register-transfer-level (RTL) based on very high speed integrated circuits hardware description language (VHDL). The performance of the design obtained via simulation showed for the real signal, a root mean squared (RMSE) value of 4.86857e-4 when compared a MATLAB equivalent model. Similarly for the imaginary signal, an RMSE value of 3.3538e-1 was obtained. The performance for the frequency spectrum showed device (number of hardware components) utilization value of 897 with RMSE value of 0.0. The RMSE value implies that there is no error between the value from the MATLAB model of the spectrum extractor subsystem and the FPGA-based design in this work. Another reason for an RMSE value of 0.0 is that the Euclidean distance between two consecutive elements is constant in the set that represents the frequency index. For the amplitude spectrum, device utilization performance yielded a value of 46,286 with an RMSE value of 3.2e-6 when compared the equivalent MATLAB model. Similarly, the power spectrum device utilization performance yielded a value of 46, 762, and an RMSE value of 2.27237e-4 when compared with the equivalent MATLAB model. Performance comparison was also made with related works; it was observed that the design in this work showed significant improvement in terms of accuracy and resource utilization.

1.0 INTRODUCTION

A spectrum analyzer (SA) is an electronic device that is used for the extraction of frequency domain information about the nature and features of a signal under observation. As such, the SA has ubiquitous application in a range of scientific, engineering, and military undertakings. The SA is a system which has its roots in the domain of digital signal processing because its operations are undergirded by the well-known fast Fourier transform (FFT) [1]. For an SA to be effective, it must continuously monitor the signal spectrum to determine if the frequency under observation is free or in use. As a result of this requirement, three design approach exist in literature

HOW TO CITE:
Yusuf, A., Dibal, P. Y., Ngene, C. U. and Okpor, J. "Design of an FPGA Based Multi-Feature Spectrum Analyzer", *Nigerian Journal of Technology*, 2026. 45(1), pp. 1 - 16. https://doi.org/10.4314/njt.2026.3713

and they include: scan-based designs, real-time FFT based designs, and scanning-FFT designs [2]. The focus of this paper is the real-time FFT based designs which will be targeted at a Field Programmable Gate Array (FPGA). Conceptually, an SA is structurally developed as shown in Figure 1.

In Figure 1, for a real-time FFT-based design targeting an FPGA, the most important component is the FPGA chip. Hence the design presented in this paper which will be highly reconfigurable will target the FPGA chip on an FPGA board as shown in Figure 1. A major advantage of this approach is that a cheap and effective SA capable of performing a variety of spectral analysis can be designed and implemented. The choice of the FPGA is hinged the fact that FPGAs have high gate counts and high speed RAMs [3]. The combination of the HLS and VHDL makes it possible to design the processor in such a way the algorithmic part of the design will be done through HLS, while register operations will be performed through VHDL.

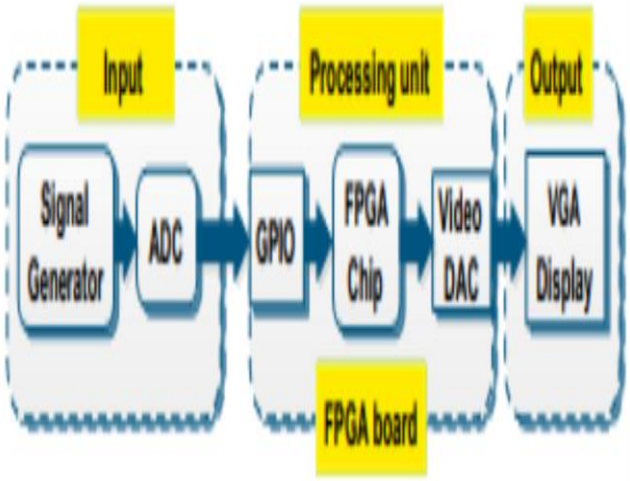


Figure 1: SA structure

In any engineering design aimed at making improvements to existing designs, it is important to conduct a review of previous designs with special focus on their limitations, and to properly understand the math undergirding the design undertaking in view. It is in this regard that the next section will present a review of previous designs of spectrum analyzers; this will be followed by a mathematical review of the math which plays a central role in the operation of spectrum analyzers.

Quite a number of design efforts for spectrum analyzers have been made in literature by different authors using a variety of methodologies. As an example, the authors in [4] proposed an SA based on a microcontroller and an FPGA. The design consisted of a controllable gain amplifier circuit, an active filter circuit, an A/D (analog-to-digital) and D/A (digital-to-analog) conversion circuits, and a main control circuit. A real-time audio spectrum analyzer was developed by the authors in [5] which analyzed loud speaker sounds; their approach involved an FFT function in calculating the amplitude of an acquired input signal which was captured by a microphone. An FFT algorithm was used by the authors in [6] in developing an FPGA-based spectrum analyzer which was designed for measuring the amplitude of an input signal. Using a technique called Yasmirub method, the authors in [2] developed a VLSI-based frequency spectrum analyzer for the extraction of the intensity factors in a signal spectrum; amplitude was the signal feature that was extracted by their technique. The authors in [7] designed a digital spectrum signal analyzer targeting an FPGA; the SA in their approach performed real-time signal analysis and real-time spectrum display. A simple SA was proposed by the authors in [8] for a signal with a bandwidth of 10MHz to 100MHz and having a resolution of 100KHz. The signal their design measured is amplitude, and the targeted hardware for their designs were AD9854, STM32F407, AD381, and AD385. A DSP Builder was used by [9] for the design and realization of real-time SA; their approach used Verilog HDL based on the Altera FFT_IP core for converting the design directly to a target FPGA.

Having reviewed previous design attempts for a SA, Table 1 shows a summary of related design attempts for a SA which have been reviewed. The strength and weakness of each design is shown, and the last row shows the expected performance of the spectrum analyzer design which will be undertaken in this work.

https://doi.org/10.4314/njt.2026.3713

Table 1: Key findings from review of related works in the design of Spectrum Analyzer

S/N	Author(s)	Title	Year	Measured signal feature				Strength	Weakness
				Power	Amplitude	Phase	Freq.		
1	M. Lv, T. Gao, D. Yan, and L. Qiao [4]	Design of Audio Signal Analyzer Based on MCU and FPGA	2020	**No**	**No**	**No**	**Yes**	The concurrent A/D conversion performed by the FPGA and the FFT operation performed by the ARM processor makes the design robust and fast.	The presence of pre-processing circuits, and multiple interfaces between the FPGA and ARM processor makes the design less optimal in the contexts of memory resources, i/o resources, and dynamic power consumption.
2	B. J. R. Raharjo and M. Zakaria [5]	Real-Time Audio Spectrum Analyzer to Analyze Loudspeaker	2019	**No**	**Yes**	**No**	**No**	The technique made it possible to determine the exact location of the audio frequency spectrum.	Slight inaccuracies were observed in the output of the spectrum analyzer when compared with the original signal. This can be a challenge in applications which have a high degree of sensitivity.
3	R. Borade, A. Dimber, D. Gharpure, and S. Ananthakrishnan [6]	Design and Development of FPGA-Based Spectrum Analyzer	2018	**No**	**Yes**	**No**	**No**	The accumulation of the FFT magnitude outputs over a windowed section caused enhanced sensitivity and accuracy.	A high number of device utilization was observed due to the memory requirements for performing the accumulation operation.
4	M. Jasmin and S. B. Hemalatha [2]	VLSI-Based Frequency Spectrum Analyzer For Low Area Chip Design By Using YASMIRUB Method	2017	**No**	**Yes**	**No**	**No**	The technique avoids memory processing operations and only requires two multipliers and three adders for real number computations.	It is not clear how accurate the technique is, and it was not established exactly what properties of a spectrum can be analyzed using this technique.

https://doi.org/10.4314/njt.2026.3713

5	D. Qi, X. Guo, and W. Du [7]	Design of Digital Signal Spectrum Analyzer Based on FPGA	2015	**No**	**Yes**	**No**	**No**	The use of CFFT algorithm in the design yielded better accuracy than FFT IP core.	A non-linear absolute error curve with varying frequency was observed in the design.
6	H. Yu, S. Kexue, L. Yanming, Z. Siqing, and C. Xiefeng [8]	A Design and Implementation of Simple Spectrum Analyzer Based on DDS	2015	**No**	**Yes**	**No**	**No**	A comparison between the spectrum chart obtained in the design and practical test showed a reasonable degree of accuracy.	The design is not reconfigurable and as such any future change will be expensive to implement.
7	M. Zengchui and W. Xin [9]	Design and Realization of Real-Time Spectrum Analysis System Based on DSP Builder	2012	**No**	**Yes**	**No**	**No**	The use of the Altera FFT_IP core ensured that an accurate and reliable FFT engine which is critical to the overall performance of the design was developed.	A very high level of precise operation must be met in order to achieve an optimal performance by the block floating point used in the FFT_IP function since it is balanceable between fixed point and floating point.
8	Proposed work		2022	**Yes**	**Yes**	**Yes**	**Yes**	The use of transaction level modeling (TLM) and register transfer level (RTL) in the design is envisaged to produce a very fast and accurate SA.	High i/o resource utilization is envisaged in the design because of the number of signal features to be determined, and the interface between the FFT stage and signal extraction stages of the processor.

https://doi.org/10.4314/njt.2026.3713

In this work, the advantage of the approach shown in Figure 1 will be taken further by combining High Level Synthesis (HLS) design using C languages, and Register Transfer Level (RTL) design using VHDL. Hence, the major contribution of this work through this combination will be a design approach based on rapid prototyping that is characterized by high level optimization, and low level precision and control.

3.0 MATHEMATICS FOR A SPECTRUM ANALYZER OPERATION

The SA design in this paper will compute four different frequency spectrum parameters as indicated in Table 1. To realize this objective, four different mathematical operations will be analyzed; these include: Fast Fourier Transform (FFT) math operations, Amplitude spectrum math operations, Phase spectrum math operations, and Power spectrum math operations.

3.1 FFT Math Operations

The FFT is a mathematical transform which converts a signal from the time domain to the frequency domain. As such, it plays a central role in signal processing, image processing, communication systems, and a host of other fields [10], [11]. Mathematically, the FFT and its inverse i.e. IFFT are defined in equation (1) and equation (2) respectively as: [1], [12], [13]

$$X(w_k) = \sum_{n=0}^{N-1} x(t_n) e^{-jw_k t_n}, 0,1,2,\ldots,N-1 \quad (1)$$

$$x(t_n) = \frac{1}{N} \sum_{k=0}^{N-1} x(w_k) e^{-jw_k t_n}, n = 0,1,2,\ldots,N-1 \quad (2)$$

where $X(w_k)$is the spectrum of x, at radian frequency w_k, $w_k \triangleq k\Omega$ is the kth frequency sample (rad/sec), $\Omega = \frac{2\pi}{NT}$is the radian-frequency sampling interval, $t_n \triangleq nT$is the nth sampling instant (sec), $n \triangleq$is the sample number (integer), N is the number of samples in both tine and frequency (integer)

In signal processing, the relationships in (1) and (2) are succinctly expressed as shown in (3) and (4).

$$X(k) \triangleq \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi nk}{N}}, k = 0, 1, 2, \ldots, N-1 \quad (3)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{\frac{j2\pi nk}{N}} \quad (4)$$

One major advantage FFT provides is the ability to see information about signal in the frequency domain which cannot be seen in the time domain. Consider the following examples as shown in Figures 2, where the FFT functions of time-domain signals are computed.

3.2 Amplitude Spectrum Math Operations

The amplitude spectrum of a signal can be derived from (3) through the following relationship [14], [15], [16]:

$$A_k = \frac{1}{N}|X(k)| = \frac{1}{N}\sqrt{(a\ l[X(k)])^2 + (a\ g[X(k)])^2}, k = 0,1,2,\ldots,N-1 \quad (5)$$

Figure 3 shows a simulation in which the left column of the figure i.e. Figures 3a-b show two periodic signals with different frequencies. Figure 3c-d show the computed amplitudes for each signal based on the relationship in (5).

3.3 Phase Spectrum Math Operations

Mathematically, the phase spectrum of a signal decomposed by the FFT can be determined using the following relationship [14], [15], [16] which is derived from (3) as follows:

$$\varphi_k = \left(\frac{a\ g[X(k)]}{a\ l[X(k)]}\right), k = 0,1,2,\ldots,N-1 \quad (6)$$

Using the relationship in 6, a simulation is performed in which the phase spectrum of two periodic signals are generated as shown in Figure 4.

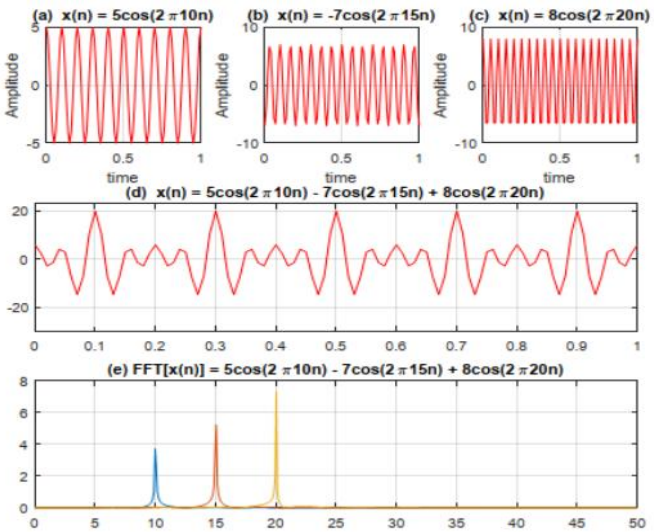


Figure 2: FFT computation of function

https://doi.org/10.4314/njt.2026.3713
http://creativecommons.org/licenses/by-nc-nd/4.0/

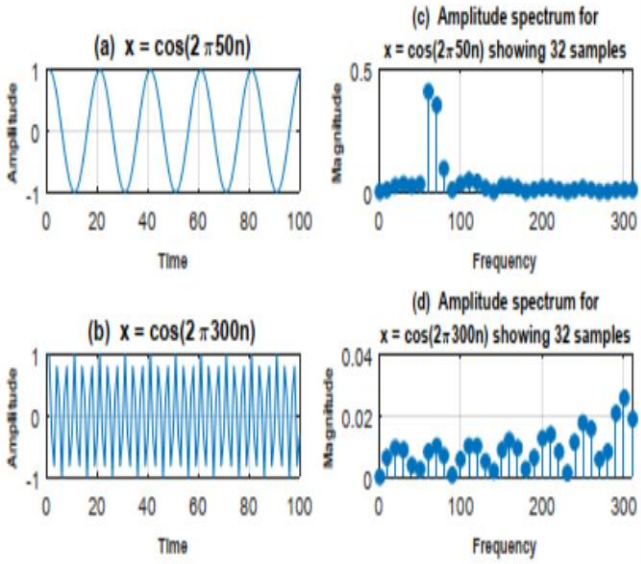


Figure 3: Amplitude spectrum of periodic signals

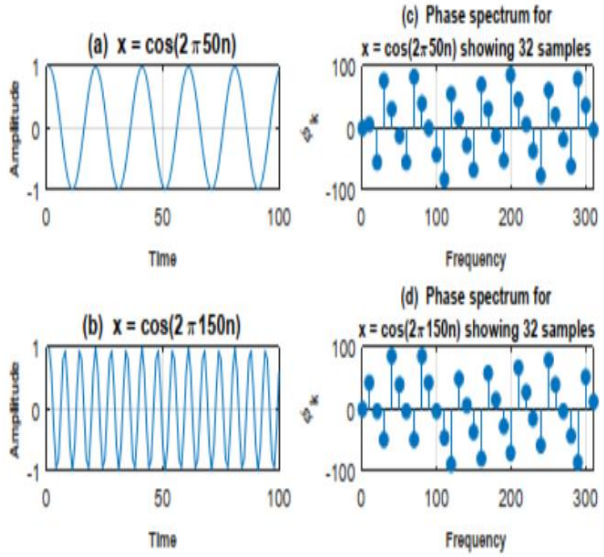


Figure 4: Phase spectrum of periodic signals

3.4 Power Spectrum Math Operations

Similar to the relationships in equations (5) and (6), the power density spectrum of a signal can be mathematically determined as [14], [15], [16] from (3) as follows:

$$P_k = \frac{1}{N^2}|X(k)|^2 = \frac{1}{N^2}[(a\,l[X(k)])^2 + (a\,g[X(k)])^2], \quad k = 0,1,2,\ldots,N = 1 \qquad (7)$$

Figure 5 shows the power density spectrum of two periodic signals which were plotted using the relationship in equation (7).

From the relationships in (5), (6), and (7), it is clear that the operation of the SA relies heavily of the relationship for the FFT of a signal as presented in (3). Conceptually, the dependence of the spectrum analyzing capability of the SA on the FFT is depicted in Figure 6.

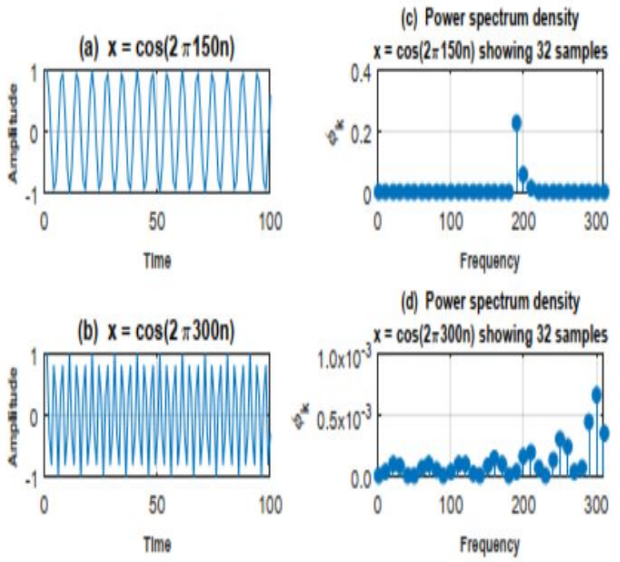


Figure 5: Power spectrum density of periodic signals

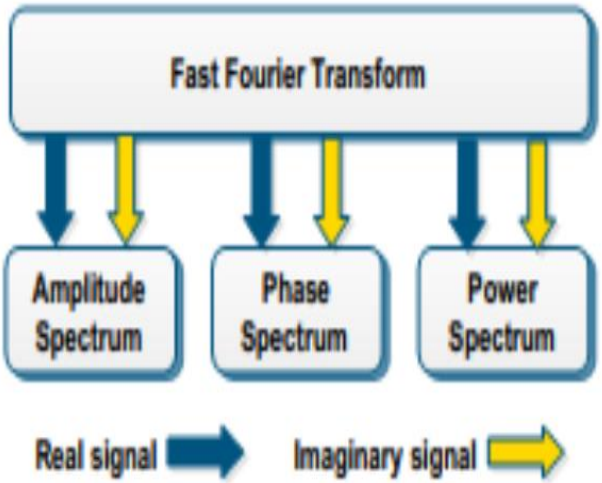


Figure 6: Dependence of SA operations on FFT

4.0 DESIGN METHODOLOGY

To realize the spectrum analyzer proposed in this on an FPGA, an architecture will be designed in such a manner that it will be able to perform FFT transform of a signal in one stage, and then perform spectrum extraction of a signal in another stage. Figure 7 shows such an architecture that is made of two sub-processors i.e. the FFT subsystem, and the spectrum

https://doi.org/10.4314/njt.2026.3713
http://creativecommons.org/licenses/by-nc-nd/4.0/

extractor subsystem. The input stage reads a file which contains the source signal to be processed by the SA, and then sends it to the data RAM when requested. The FFT subsystem receives signal from the data RAM and then performs an FFT transform that yields a real and imaginary component of the signal. These signals are then transmitted to the spectrum extractor subsystem which uses them to compute the corresponding frequency index, amplitude spectrum, phase spectrum, and power spectrum. Communication between different subsystems in the design is performed using the AXI stream protocol; the reader is referred to [17] for further information on this protocol.

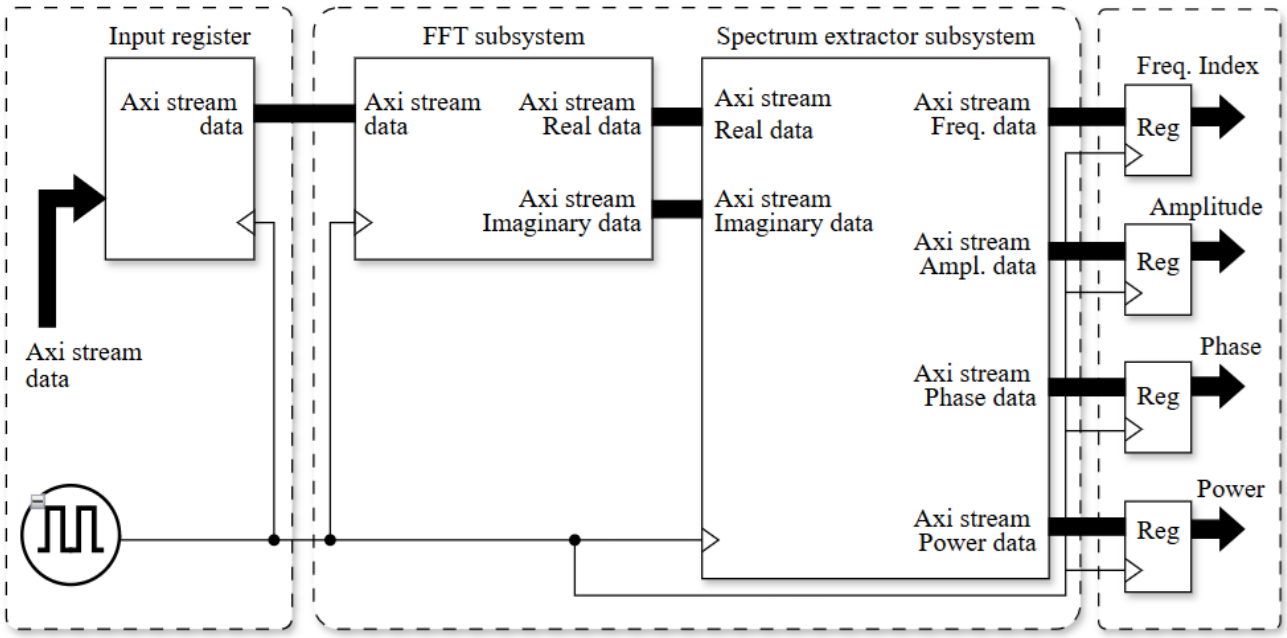


Figure 7: Spectrum analyzer architecture

4.1 FFT Subsystem Architecture Design

The architecture for the FFT considered in this work which is based on the decimation in frequency (DIF) shown in Figure 8 for a case 16 samples and corresponding 4 iteration levels for the sake of brevity. The reader is referred to [16], [18], [19], [20] for more information on DIF technique. A High Level Synthesis (HLS) approach will be used in coding the architecture in Figure 3.3 where the C language and data types will be used directing the signal flow and creating the required interfaces. Also, the math library in C will be used in computing the sine and cosine of signal angles as shown in Figure 8. The flow of signals from the input to the output in the context of indexes is shown in Figure 9 where signals in the first iteration use 1 as superscript, signals in the second iteration use 2 as superscript, signals in the third iteration use 3 as superscript; superscripts are not used in the last iteration since that is where the output is expected. It should be noted that the position of an output index in the DIF architecture is derived using bit reversal technique.

4.2 Spectrum Extractor Subsystem Architecture Design

The architecture for the spectrum extractor subsystem is shown in Figure 10 where the real and imaginary signals from the FFT subsystem, the sampling frequency, and number of samples are used as input. The output of the spectrum extractor includes the frequency index, amplitude, phase, and power spectrums. The basis for these outputs are defined in equation 5 – 7; the definition of the mathematical operations performed on the real and imaginary parts of the signal was used in designing the architecture for the spectrum extractor subsystem. Similar to the FFT subsystem, the design implementation for the spectrum extractor subsystem will be performed using HLS and C as the coding language. The rich array of functions in the math library of C will be used in performing the required arithmetic operations for spectrum extraction.

https://doi.org/10.4314/njt.2026.3713
http://creativecommons.org/licenses/by-nc-nd/4.0/

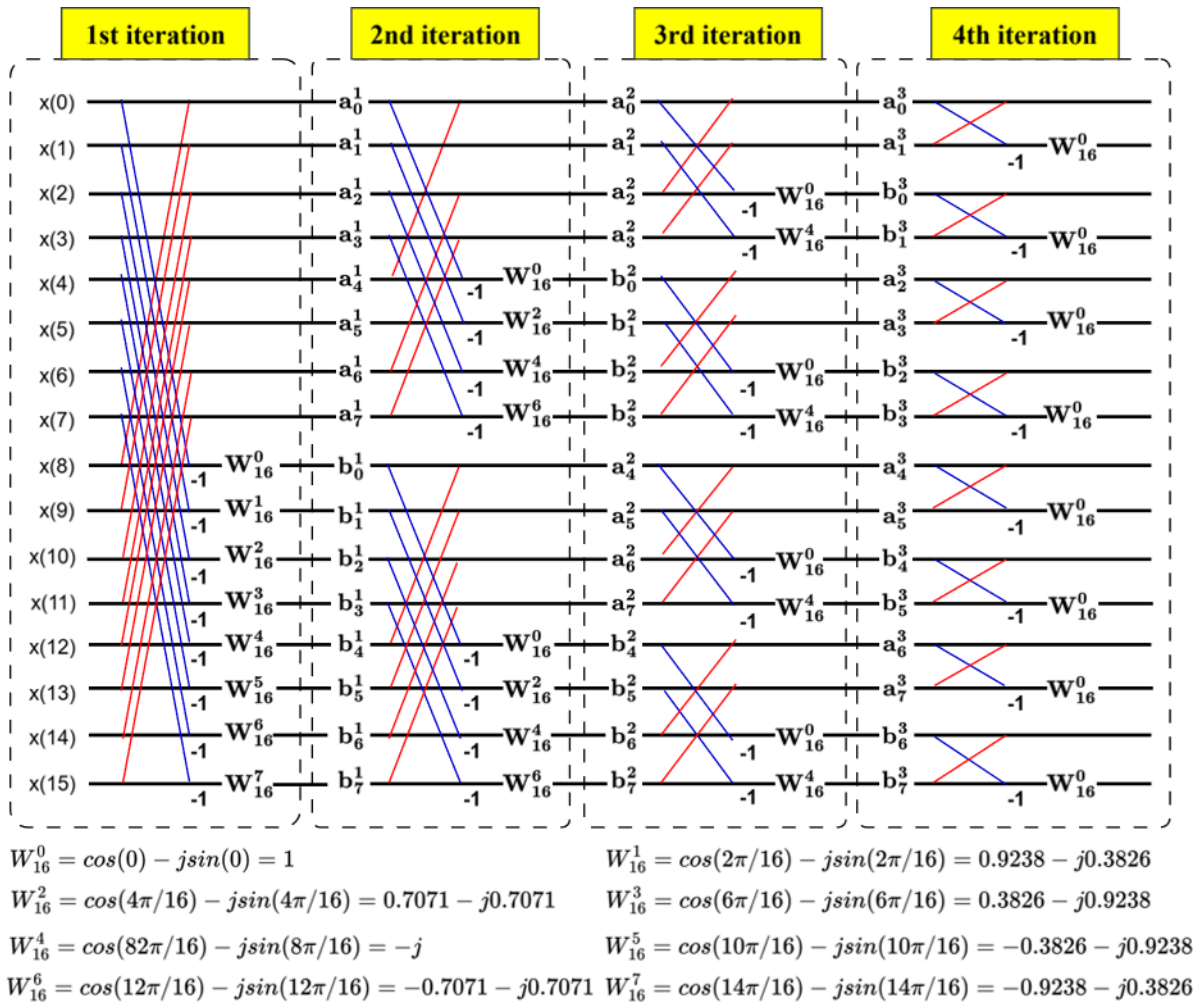


Figure 8: DIF-based FFT architecture

5.0 ANALYSIS OF SIMULATION RESULTS AND PERFORMANCE COMPARISON

The input stage of the design as shown in Figure 7 is where the data file is read and sent to the RAM before being accessed by the FFT subsystem. Table 2 shows the data that was stored in the file which was read in the input stage.

The reading of the data by the input stage is shown in Figure 11 where for the sake of brevity, the first eight samples that were read are shown. The data from the input stage are fed to the FFT subsystem through the RAM; the FFT subsystem performs an FFT operation according to equation 1. As indicated in Figure 7, two outputs i.e. the real signal and the imaginary are output by the FFT subsystem. Figure 12 shows four samples of results that were output by the FFT subsystem in this work.

To validate the accuracy of the FFT subsystem, the FFT transform of the signal from the input stage is tested with the standard FFT function in MATLAB. Table 3 shows the results obtained from the standard FFT function in MATLAB, and the FFT subsystem in this paper.

Table 2: Data stored in file read by the input stage

S/N	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Input Data	-1.6642	-0.59	-0.2781	0.4227	-1.6702	0.4716	-1.2128	0.0662
S/N	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Input Data	0.6524	0.3271	1.0826	1.0061	-0.6509	0.2571	-0.9444	-1.3218

https://doi.org/10.4314/njt.2026.3713
http://creativecommons.org/licenses/by-nc-nd/4.0/

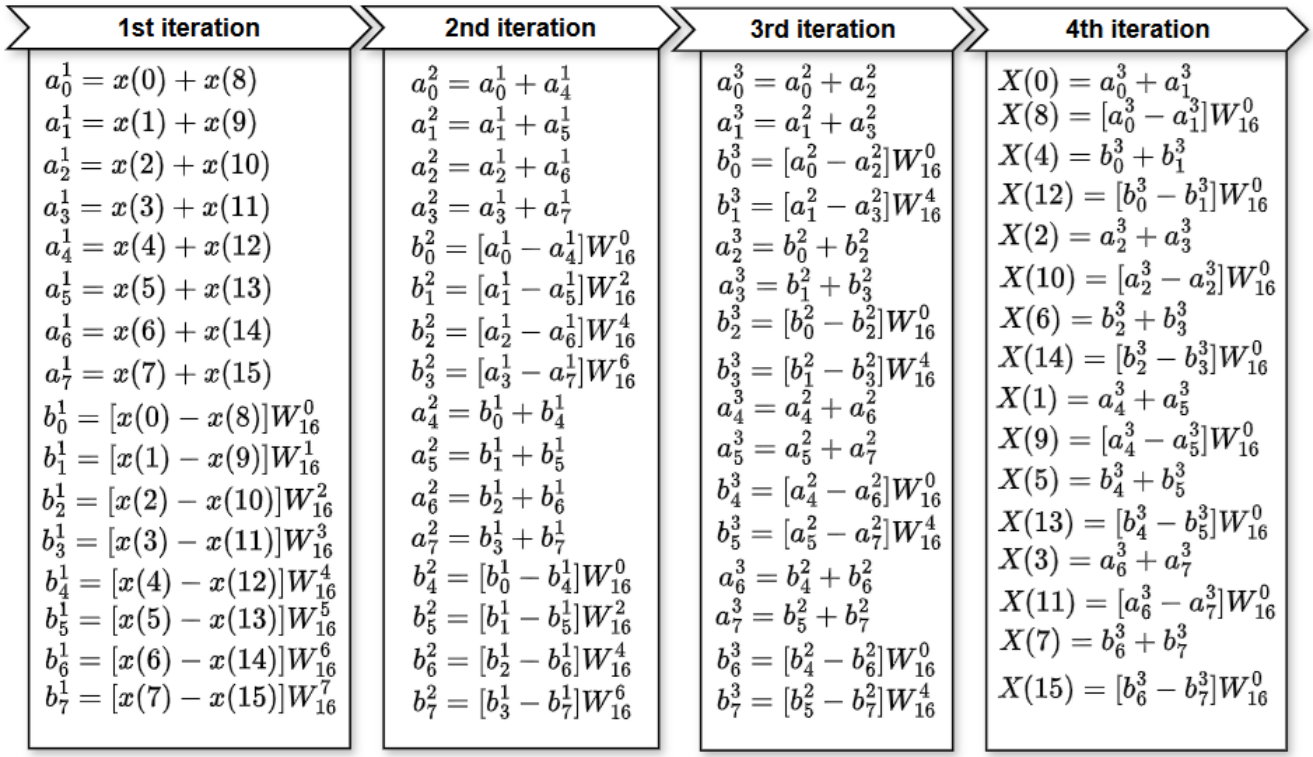


Figure 9: Indexed signal flow in DIF architecture

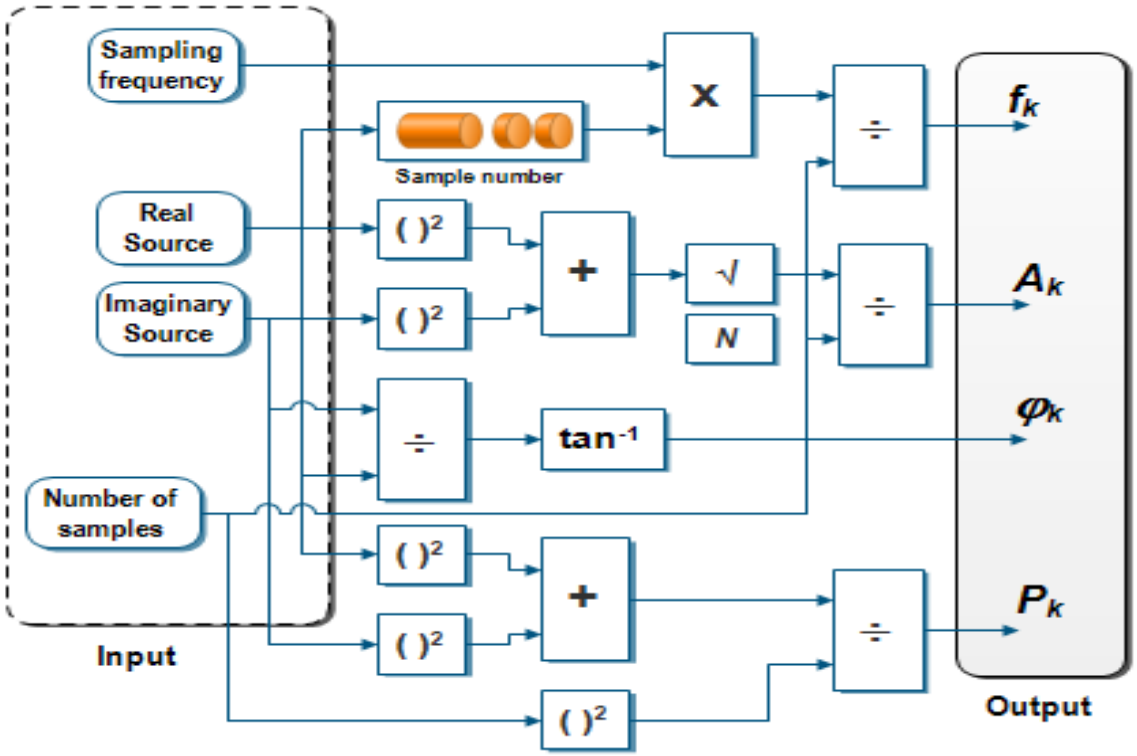
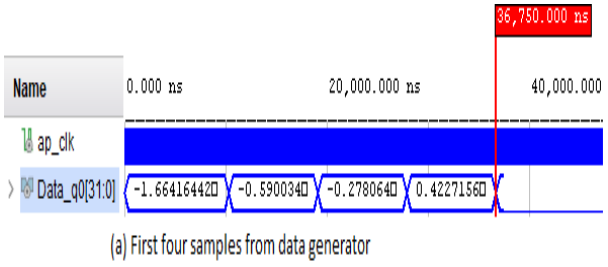


Figure 10: Spectrum extractor architecture

https://doi.org/10.4314/njt.2026.3713
http://creativecommons.org/licenses/by-nc-nd/4.0/



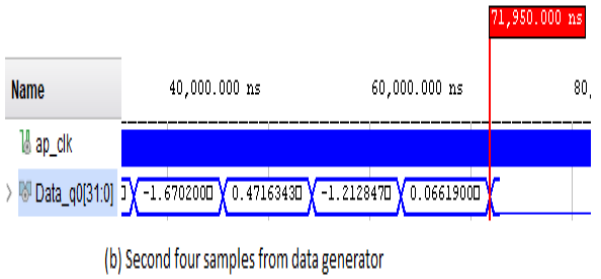
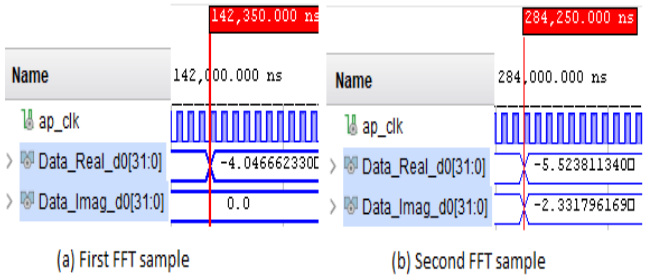


Figure 11: Reading of data in the input stage



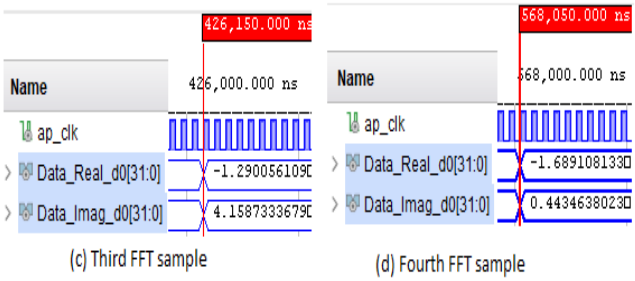


Figure 12: Four samples of FFT subsystem output

Table 3: FFT computation using MATLAB and FFT subsystem

S/N	Input Data	FFT Data (MATLAB)		FFT Data (VHDL)		Error (Real)	Error (Imaginary)
		Real	Imaginary	Real	Imaginary		
0	-1.6642	-4.046660	0.000000	-4.046662	0.000000	0.000002	0.000000
1	-0.5900	-5.523869	2.331754	-5.525811	-2.331796	0.001942	-0.000042
2	-0.2781	-1.290118	-4.158689	-1.290056	4.158733	-0.000062	0.000044
3	0.4227	-1.689136	-0.443549	-1.689108	0.443463	-0.000028	-0.000086
4	-1.6702	-1.980310	-0.292510	-1.980256	0.292521	-0.000054	0.000011
5	0.4716	-1.399323	-0.708836	-1.399275	0.708866	-0.000048	0.000030
6	-1.2128	3.908658	1.764751	3.908659	-1.764877	-0.000001	-0.000126
7	0.0662	-0.653912	-2.010693	-0.653874	2.010771	-0.000038	0.000078
8	0.6524	-5.324520	0.000000	-5.324483	1.341553	-0.000037	1.341553
9	0.3271	-0.653912	2.010693	-0.653873	-2.010782	-0.000039	-0.000089
10	1.0826	3.908658	-1.764751	3.908673	1.764819	-0.000015	0.000068
11	1.0061	-1.399323	0.708836	-1.399307	-0.708876	-0.000016	-0.000040
12	-0.6509	-1.980310	0.292510	-1.980276	-0.292522	-0.000034	-0.000012
13	0.2571	-1.689136	0.443549	-1.689127	-0.443471	-0.000009	0.000078
14	-0.9444	-1.290118	4.158689	-1.290122	-4.158726	0.000004	-0.000037
15	-1.3218	-5.523869	-2.331754	-5.523803	2.331852	-0.000066	0.000098

The last two columns of Table 3 shows error values that were determined as the difference between the values obtained from the MATLAB and FFT subsystem for FFT computations. To validate the error analysis for the real and imaginary, a root mean square error (RMSE) analysis was performed using the relationship in (8) as follows [21]:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \left(y(i)-\hat{y}(i)\right)^2}{N}} \quad (8)$$

where $y(i)$ is the output (obtained from MATLAB simulation), $\hat{y}(i)$ is the actual output (obtained from the VHDL-based hardware description), and *N* is the number of samples. The RMSE results obtained are RMSE (Real) = 4.86857e-4 and RMSE (Imaginary) = 3.3538e-1

The simulation for the spectrum extractor that was designed based on the architecture in Figure 10, is show in Figure 13 for first two samples of the other outputs of the spectrum extractor. Table 4 shows

https://doi.org/10.4314/njt.2026.3713

error analysis for the frequency spectrum and the amplitude spectrum. The error was computed from the values obtained from MATLAB and the values obtained from the spectrum extractor. Using the relationship in equation 8, the RMSE values for the frequency and the amplitude spectrum were determined as RMSE (Frequency spectrum) = 0.0000 and RMSE (Amplitude spectrum) = 3.20e-6 respectively. The reason for an RMSE value of 0.0000 for the frequency spectrum is that the Euclidean distance is constant for any two consecutive elements in the set that makes up the frequency index column in Table 4 for the AMTLAB model and the spectrum extractor sub-system. The value of the Euclidean distance is determined as 6.25 for any two consecutive elements as indicated.

Table 5 shows the error analysis for the phase and the power spectrum. In a similar error analysis for the phase and power spectrum, the RMSE values were obtained as RMSE (Phase spectrum) = 6.2991e-1, and RMSE (Power spectrum) = 2.27237e-4.

A performance comparison was made between the outputs of the spectrum analyzer in this work and the results obtained from similar works. As shown in Figure 13, the first output of the spectrum extractor is the frequency; consequently, Table 6 shows the performance of the spectrum extractor for computation of frequency in comparison to similar works where it can be seen that the design in this work showed much better accuracy, and significant improvement in efficient resource utilization. On the aggregate, 962690 devices were used by [22], 253918 devices were used by [23], 11042 devices were used by [7], while the design in this work used 897 devices.

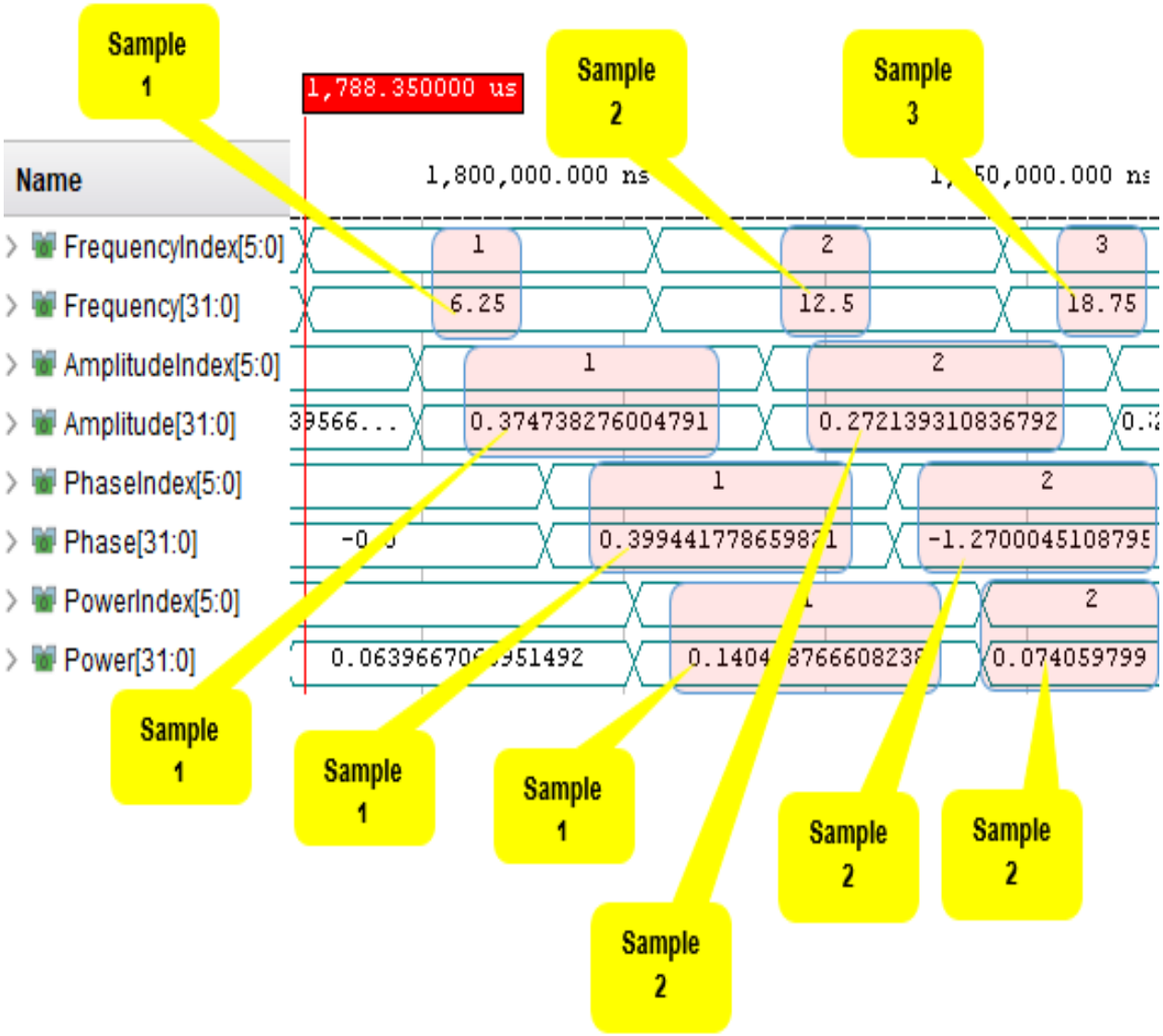


Figure 13: First two to three samples of spectrum extractor

https://doi.org/10.4314/njt.2026.3713
http://creativecommons.org/licenses/by-nc-nd/4.0/

Table 4: Error analysis for frequency and amplitude spectrum

S/N	Freq Index (MATLAB)	Freq Index (Spectrum Extractor)	Error	Amplitude (MATLAB)	Amplitude (Spectrum Extractor)	Error
0	00.00	00.00	00.00	0.25290969	0.25291639	-0.000006700
1	06.25	06.25	00.00	0.37474888	0.37473827	0.000010610
2	12.50	12.50	00.00	0.27213071	0.27213931	-0.000008600
3	18.75	18.75	00.00	0.10915611	0.10914701	0.000009100
4	25.00	25.00	00.00	0.12510490	0.12510906	-0.000004160
5	31.25	31.25	00.00	0.09804007	0.09803663	0.000003440
6	37.50	37.50	00.00	0.26804171	0.26803982	0.000001890
7	43.75	43.75	00.00	0.13214903	0.13215099	-0.000001960
8	50.00	50.00	00.00	0.33278531	0.33278021	0.000005100
9	56.25	56.25	00.00	0.13214903	0.13215158	-0.000002550
10	62.50	62.50	00.00	0.26804171	0.26803913	0.000002580
11	68.75	68.75	00.00	0.09804007	0.09803871	0.000001360
12	75.00	75.00	00.00	0.12510490	0.12511031	-0.000005410
13	81.25	81.25	00.00	0.10915611	0.10914830	0.000007810
14	87.50	87.50	00.00	0.27213071	0.27214011	-0.000009400
15	93.75	93.75	00.00	0.37474888	0.37473917	0.000009710

Table 5: Error analysis for phase and power spectrum

S/N	Phase (MATLAB)	Phase (Spectrum Extractor)	Error	Power (MATLAB)	Power (Spectrum Extractor)	Error
0	0.00000000	0.00000000	0.00000000	0.06396331	0.0639667	-0.00000339
1	-0.39943375	0.39944177	0.00000802	0.14043672	0.1404288	0.00000796
2	1.26999514	-1.27000451	-0.00000937	0.07405512	0.0740598	-0.00000467
3	0.25678422	-0.25674870	0.00003552	0.01191506	0.0110131	0.00090199
4	0.14666611	-0.14665822	0.00000789	0.01565124	0.0156523	-0.00000103
5	0.46892541	-0.46890988	0.00001553	0.00961186	0.0096112	0.00000068
6	0.42409191	-0.42412573	-0.00003382	0.07184636	0.0718454	0.00000101
7	1.25632355	-1.25639629	-0.00007274	0.01746337	0.0174639	-0.00000051
8	0.00000000	-2.51959318	-2.51959318	0.11074606	0.1107427	0.00000339
9	-1.25632355	-1.25639820	0.00007465	0.01746337	0.017464	-0.00000067
10	-0.42409191	0.42411205	0.00002014	0.07184636	0.071845	0.00000139
11	-0.46892541	0.46890661	-0.00001880	0.00961186	0.0096116	0.00000028
12	-0.14666611	0.14665748	-0.00000863	0.01565124	0.0156526	-0.00000134
13	-0.25678422	0.25675022	-0.00003400	0.01191506	0.0119134	0.00000171
14	-1.26999514	1.26998949	-0.00000565	0.07405512	0.0740602	-0.00000512
15	0.39943375	-0.39945095	-0.00001720	0.14043672	0.1404295	0.00000727

https://doi.org/10.4314/njt.2026.3713
http://creativecommons.org/licenses/by-nc-nd/4.0/

Table 6: Performance comparison for spectrum extractor for frequency spectrum operation

S/N		1	2	3	4	5	6
Author(s)		A. Sarma, *et al* [22]	M. Lv, *T et al* [4]	Tong *et al* [23]	R. Borade *et al* [6]	D. Qi *at al* [7]	Design in this work
Year		2021	2020	2020	2018	2015	2023
Title of Article		Design and Implementation of a Re-configurable Wideband Radio Frequency Spectrum Analyzer with Image Rejection in the Digital Domain	Design of Audio Signal Analyzer Based on MCU and FPGA	A high performance fast-Fourier-transform spectrum analyzer for measuring spin noise spectrums	Design and Development of FPGA-Based Spectrum Analyzer	Design of Digital Signal Spectrum Analyzer Based on FPGA	Design of an FPGA Based Multi-Feature Spectrum Analyzer
RMSE	**Frequency**	N/R	0.005307	N/R	0.040000	0.011016	0
Device Utilization	**Slice Regs**	26,660	N/R	44,601	4,712	N/R	237
	Slice LUTs	39,126	N/R	31,281	6,285	N/R	277
	IOBs	896,672	N/R	840	28	N/R	0
	DSP48	232	N/R	1,300	17	N/R	3
	Flip flops	N/R	N/R	175,896	N/R	N/R	380

N/R = Not Reported

https://doi.org/10.4314/njt.2026.3713

Table 7: Performance comparison for SA amplitude spectrum operation

Compared parameter		**Amplitude spectrum comparison**		**Power spectrum comparison**	
S/N		1	2	1	2
Author(s), year		B. M. L. Vigil, *et al* [25]	Design in this work	Flak, P., [24]	Design in this work
Year		2013	2023	2022	2023
Title of Article		FPGA Based Spectrum Analyzer	Design of an FPGA Based Multi-Feature Spectrum Analyzer	Hardware-Accelerated Real-Time Spectrum Analyzer With a Broadband Fast Sweep Feature Based on the Cost-Effective SDR Platform	Design of an FPGA Based Multi-Feature Spectrum Analyzer
RMSE	**Power**	N/R	0.00000320	4.879415	0.00022723
Device Utilization	**Slice Regs**	N/R	839	N/R	863
	Slice LUTs	21,753	17,928	N/R	18,356
	IOBs	N/R	60	N/R	60
	DSP48	29	93	N/R	93
	Flip flops	79,404	27,366	N/R	27,390

N/R = Not Reported

https://doi.org/10.4314/njt.2026.3713

For amplitude spectrum computation, Table 7 shows the performance of the design in this work where a much better accuracy, and efficient resource utilization was observed. The aggregate performance shows 101,186 devices were used by Vigil *et al*, while the design in this work used 46,286 devices. The performance comparison for the power spectrum computation is also shown in Table 7 where it can be seen that the accuracy of the result in the design in this paper is much better than work by [24]. It should be noted that performance comparison for phase spectrum computation was not made; we have not been able to come across a literature that focused phase spectrum

CONCLUSION

This paper presented the design of a spectrum analyzer capable of computing four critical spectrum features- frequency, amplitude, phase, and power. The design in this work extends the capabilities of spectrum analyzers in literature by being able to extract four features of a signal spectrum- this is an improvement on other spectrum analyzers that compute only one spectrum feature. The spectrum analyzer in this paper was divided into an FFT subsystem, and a spectrum extractor subsystem. The FFT subsystem computed the FFT of an input signal to produce a real and imaginary corresponding values, while the spectrum extractor computed four spectrum features as earlier mentioned. A performance comparison was made with related work in literature; it is shown that for all the criteria considered i.e. RMSE and device utilization in Tables 6 and 7, the design in this work showed a much better performance.

REFERENCES

[1] Smith, J. O., III, "Introduction to the DFT," in *Mathematics of the Discrete Fourier Transform (DFT)*, 2nd ed., Stanford, California: CCRMA, 2002. chapter 1, pp. 1–3. [Online]. Available: http://ccrma.stanford.edu/~jos/mdft/

[2] Jasmin, M. and Hemalatha, S, B. "VLSI-Based Frequency Spectrum Analyzer for Low Area Chip Design by using Yasmirub Method," *Int. J. Pure Appl. Math.*, 116(15), pp. 557–561, 2017.

[3] Ashenden, J, P. *Digital Design: An Embedded Systems Approach Using VHDL*. Burlington, MA: Morgan Kaufmann, 2008. [Online]. Available: https://shop.elsevier.com/books/digital-design-vhdl/ashenden/978-0-12-369528-4

[4] Lv, M. Gao, T. Yan, D. and Qiao, L. "Design of Audio Signal Analyzer Based on MCU and FPGA," *J. Physical Conference Ser.*, 1544 (1), pp. 1–6, 2020. doi: 10.1088/1742-6596/1544/1/012111.

[5] Raharjo, B, J, R. and Zakaria, M. "Real-Time Audio Spectrum Analyzer to Analyze Loudspeaker," *J. Physical Conference Service*, 1413, pp. 1–8, 2019, doi: 10.1088/1742-6596/1413/1/012005.

[6] Borade, R. Dimber, A. Gharpure, D. and Ananthakrishnan, S. "Design and Development of FPGA-Based Spectrum Analyzer," *IETE J. Educ.*, 59(1), pp. 5–17, 2018. doi: 10.1080/09747338.2018.1450648.

[7] Qi, D. Guo, X. and Du, W. "Design of Digital Signal Spectrum Analyzer Based on FPGA," in *Proceeding of the 2015 IEEE International Conference on Information and Automation*, Lijiang, China, 2015. pp. 2725–2729.

[8] Yu, H. Kexue, S. Yanming, L. Siqing, Z. and Xiefeng, C. "A Design and Implementation of Simple Spectrum Analyzer Based on DDS," in *3rd International Conference on Machinery, Materials and Information Technology Applications*, Qingdao, China, 2015. pp. 1257–1261. doi: 10.2991/icmmita-15.2015.232

[9] Zengchui, M. and Xin, W. "Design and Realization of Real-Time Spectrum Analysis System Based on DSP Builder," in *Second International Conference on Instrumentation & Measurement, Computer, Communication and Control*, Harbin, china, 2012. pp. 1077–1080. doi: 10.1109/IMCCC.2012.254.

[10] Awah, C, Amaghionyeodiwe, C, A. Obasi, O. Oti, S, E. and Nnabuenyi, I, K. "Electromechanical Impact of Poles on the Performance of Double Stator," *Nigeria Journal Technology*, 43(3), pp. 534–541, 2024. doi: 10.4314/njt.v43i3.15

[11] Onovo, H, *et al.* "Properties Enhancement and Compositional Optimization Study of Tailored Nanosilica Reinforced Bioplastic Film Composites," *Nigeria Journal Technology*, 44(1), pp. 17–28, 2025. doi: 10.4314/njt.v44i1.3.

[12] Chu, E. "DFT and Band-Limited Periodic Signal," in *Discrete and Continuous Fourier Transforms: Analysis, Applications and Fast Algorithms*, Boca Raton: CRC Press, 2008. chapter 2, p. 29.

[13] Broughton, S, A. and Bryan, K. "The One-Dimensional DFT," in *Discrete Fourier Analysis and Wavelets: Applications to Signal and Image Processing*, 2nd ed., Hoboken, New Jersey: Wiley, 2018. chapter 2, p. 79.

[14] Cerna, M. and Harvey, A, F. "The Fundamentals of FFT- Based Signal Analysis and Measurement," National Instruments Application Note 041. Accessed: 2022. [Online]. Available: https://www.sjsu.edu/people/burford.furman/docs/me120/FFT_tutorial_NI.pdf

[15] CENG, "The Discrete Fourier Transform and the FFT." Accessed: Apr. 25, 2022. [Online]. Available: https://sceweb.sce.uhcl.edu/harman/ceng5431/2015WEB/Chap11tlhDFT.pdf

[16] Tan, L. "Discrete Fourier Transform and Signal Spectrum," in *Digital Signal Processing: Fundamentals and Applications*, London, UK: Academic Press, 2008. chapter 4, pp. 87–131. doi: 978-0-12-374090-8.

[17] Arm Ltd, "AMBA AXI and ACE Protocol Specification," 2022. *Arm Ltd*. [Online]. Available: https://developer.arm.com/documentation/ihi0022/e/

[18] Ingle, K, V. and Proakis, J, G. *Digital Signal Processing using MATLAB*, 3rd ed. Stamford CT: Cengage Learning, 2010.

[19] M. Giron-Sierra, J, *Digital Signal Processing with MATLAB Examples*. Singapore: Springer, 2017. doi: 10.1007/978-981-10-2537-2.

[20] Ingle, V, K. and Proakis, J. G "The Discrete-Time Fourier Analysis," in *Digital Signal Processing Using MATLAB*, 3rd ed., Cengage Learning, chapter 3, Boston, MA, pp. 59–102, 2012.

[21] Hodson, T, O. "Root mean square error (RMSE) or mean absolute error (MAE): when to use them or not," *Geosci. Model Dev.*, 15 (14) pp. 1–10, 2022. doi: 10.5194/gmd-2022-64.

[22] Sarma, A. Sun, Y. Simpson, O. and Lauder, D. "Design and Implementation of a Reconfigurable Wideband Radio Frequency Spectrum Analyzer with Image Rejection in the Digital Domain," *IEEE Trans. Instrument Measure*, 70, pp. 1–16, 2021. doi: 10.1109/TIM.2021.3060578.

[23] Tong, Y, *et al*. "A high performance Fast-Fourier-transform spectrum analyzer for measuring spin noise spectrums," *Chinese Phys. Soc. IOP Publ.*, 29(9), pp. 090704-1-090704–8, 2020. doi: 10.1088/1674-1056/ab9c04.

[24] Flak, P. "Hardware-Accelerated Real-Time Spectrum Analyzer with a Broadband Fast Sweep Feature Based on the Cost-Effective SDR Platform," *IEEE Access*, 10, pp. 110934–110946, 2022. doi: 10.1109/ACCESS.2022.3215800.

[25] Vigil, B. M, L. Urquiaga, W, F. Barbolla, D, B. and Alonso, S, G. "FPGA Based Spectrum Analyzer," in *Congreso Argentino de Sistemas Embedidos*, Buenos Aires, Argentina, 2013. pp. 34–38. [Online]. Available: https://www.sase.com.art