Circuits and Systems, 2018, 9, 148-167
http://www.scirp.org/journal/cs
ISSN Online: 2153-1293
ISSN Print: 2153-1285

Peak Detection Implementation for Real-Time Signal Analysis Based on FPGA

Alperen Mustafa Colak[1], Taito Manabe[1], Yuichiro Shibata[1], Fujio Kurokawa[2]

[1]Graduate School of Engineering, Nagasaki University, Nagasaki, Japan
[2]Nagasaki Institute of Applied Science, Nagasaki, Japan
Email: colak@pca.cis.nagasaki-u.ac.jp, manabe@pca.cis.nagasaki-u.ac.jp, shibata@cis.nagasaki-u.ac.jp, kurokawa_fujio@nias.ac.jp

How to cite this paper: Colak, A.M., Manabe, T., Shibata, Y. and Kurokawa, F. (2018) Peak Detection Implementation for Real-Time Signal Analysis Based on FPGA. *Circuits and Systems*, **9**, 148-167.
https://doi.org/10.4236/cs.2018.910016

Received: October 3, 2018
Accepted: October 27, 2018
Published: October 31, 2018

Abstract

In this paper a real-time peak detection method based on modified Automatic Multiscale Field Detection (AMPD) algorithm and Field Programmable Gate Arrays (FPGA) technologies of a time series data is studied, and optimum scaling is highlighted after testing several scales. To validate the results obtained from modified algorithm, they are compared with the results of original AMPD method. As data of this study, three-phase voltage values of a power station are used. A detail detective sensitivity analysis of phase-to-phase voltage values is tried at different scales. Moreover, the original algorithm is tested regarding the off-line mode to obtain optimum scaling for real-time peak point detection. It is concluded that the peak detection of minimum and maximum points of data series achieved by modified algorithm is very close to the results of original AMPD algorithm.

Keywords

AMPD Algorithm, Off-Line Method, FPGA, Peak Detection

1. Introduction

Peak detection of any time series data is always a hot topic in many engineering fields including chemistry, biology, biomedical, optics, astrophysics and energy systems. So these fields often require real-time peak detection. As the environment noises can affect the signals somehow, a robust peak detection, in this case, is a challenging topic. To obtain a successful peak detection method, several methods have been proposed, including automatic multiscale-based peak detection [1], window-threshold techniques [2] [3] [4], wavelet transform [5]-[11], techniques using entropy [12], and artificial neural networks [13] [14]. Particularly, each method was investigated in terms of the detection method employed and the detection performance achieved. Drawbacks of the peak detection algorithms available in the literature are that many free parameters such as the window length of a threshold value have to be used in order to apply the algorithm to the signal, and to make the algorithm applicable. Generally, the algorithms with fewer parameters are restricted for use in specific applications like the detection of R-peaks in electroencephalography (ECG) signals and to obtain an adaptive and time-efficient R-peak detection algorithm for ECG processing as well as reduce the size and noise of ECG signals [15]-[20]. In addition, noise in analyzed signal is a challenge for many peak detection algorithms.

On the other hand, periodic and quasi-periodic signals are the most difficult ones to detect the peak points. However, AMPD method is suitable for all types of peak point detecting. Thus, the automatic multiscale-based peak detection (AMPD) method [1] has been introduced as an effective method. In this research, we used the off-line and online terms to show the scale. Off-line algorithm means to fix the scale, then analysis all input data with using stable scale for each clock. On the other hand, online algorithm means to vary the scale for each clock. However, these software methods are not suited for real-time processing but emphasized off-line sophisticated data analysis.

The use of FPGAs provides a promising approach to real-time peak analysis [21] [22]. FPGAs do not run a program stored in the program memory because they are reprogrammable chips and include a lot of logic gates, which are internally connected to form a complex digital circuitry. FPGAs are not processors and are entirely different from CPUs, GPUs, and DSP. However, they offer various opportunities for efficient real-time signal processing by making the best use of pipelined structure in computing. Furthermore, in previous work, we applied AMPD method on an FPGA as off-line by changing its bite size and scales and then analyzed it in terms of speed, cost and memory [23]. Eventually, we realized that changes in bit size did not affect the peak detection.

This paper introduces a novel approach for robust and real-time peak detection by using the AMPD algorithm and the FPGA technology. It highlights the modification of the original AMPD algorithm to be an off-line method, and how it can be implemented on an FPGA so that a pipelined structure in computing is extracted on hardware. Thus, the optimum scaling for the off-line peak detection is obtained, and results compared with the original AMPD method are found very promising.

2. Overview of Algorithm

2.1. AMPD Method

The AMPD method [1] is a technique especially to find the peak points of periodic and quasi-periodic noisy signals on-line. It calculates all input data by using matrix equation to find the peak points. Also, AMPD determines the local maxima points in different time periods, each of which is named scale and is the number of compared data at a time. AMPD is divided into 4 different stages: Local Maxima Scalogram (LMS) calculation, Row-Wise summation of the LMS, LMS rescaling, and peak detection. Figure 1 shows the flowchart about the original algorithm. However, to enable off-line processing on an FPGA, Row-Wise summations of the LMS and LMS rescaling are skipped to obtain optimal scale decision and LMS rescaling, which is possible to perform in advance in a calibration phase. Number of combinations to create scale pattern like 33, 65, 129, 257, 513 and 1025 for make a simple and efficient pipeline to implement on FPGA. Fixed scales are then chosen and, based on them, the sensitivities of the peak points are analyzed.

2.1.1. LMS Calculation

Essentially, LMS calculation means analysis of all input values ($x = x_1, x_2, x_3, x_4, \cdots, x_n$) by using the moving window approach to fill in the Z matrix given in Equation (1) below that also indicates the size of the Z matrix. In Z matrix, k denotes the number of rows while i denotes the columns. The relation between n and L can be defined as shown in Equation (1):

$$Z = \begin{bmatrix} z_{1,1} & z_{1,2} & z_{1,3} & \cdots & z_{1,n} \\ z_{2,1} & z_{2,2} & z_{2,3} & \cdots & z_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_{L,1} & z_{L,2} & z_{L,3} & \cdots & z_{L,n} \end{bmatrix} = \left(z_{k,i} \right) \tag{1}$$

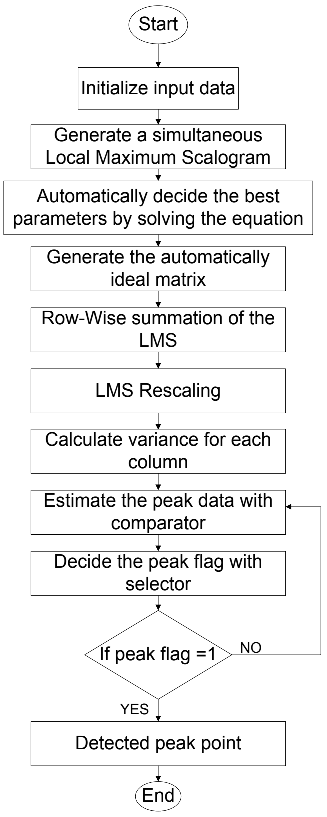

Start
Initialize input data
Generate a simultaneous Local Maximum Scalogram
Automatically decide the best parameters by solving the equation
Generate the automatically ideal matrix
Row-Wise summation of the LMS
LMS Rescaling
Calculate variance for each column
Estimate the peak data with comparator
Decide the peak flag with selector
If peak flag =1
NO
YES
Detected peak point
End

Figure 1. AMPD algorithm flowchart.

When all numbers denoted by x_i are analyzed, the previous value should be (X_{i-1}) and the next value (X_{i+1}) is checked and compared using the window approach, which is called distance between all the points. Furthermore, the window approach scale depends on the L.

where;

n denotes total column number

L denotes total row number

$$L = (n/2) - 1 \tag{2}$$

At the same time k is changed from 1 to L in Figure 2, which shows the comparison mechanism of input values.

Elements of Z matrix can be calculated by Equation (3).

$$z_{k,i} = \begin{cases} 0, & \text{if } (x_{i-1} > x_{i-k-1}) \wedge (x_{i-1} > x_{i+k-1}) \\ r, & \text{otherwise} \end{cases} \tag{3}$$

where

r denotes random number between 1 and 2.

If this condition point (x_i) is provided,

$x_i > x_{i-1}$ and $x_i > x_{i+1}$,

Then, new diagonal element $z_{L,n}$ value assigned to $z_{k,i}$ is zero in the Z matrix, or else, a random number (r) assigned to $z_{k,i}$ is generated at every time by a random generator. The range of random numbers is between $1 < r < 2$. In this way, the whole Z matrix is obtained by analyzing each element of it.

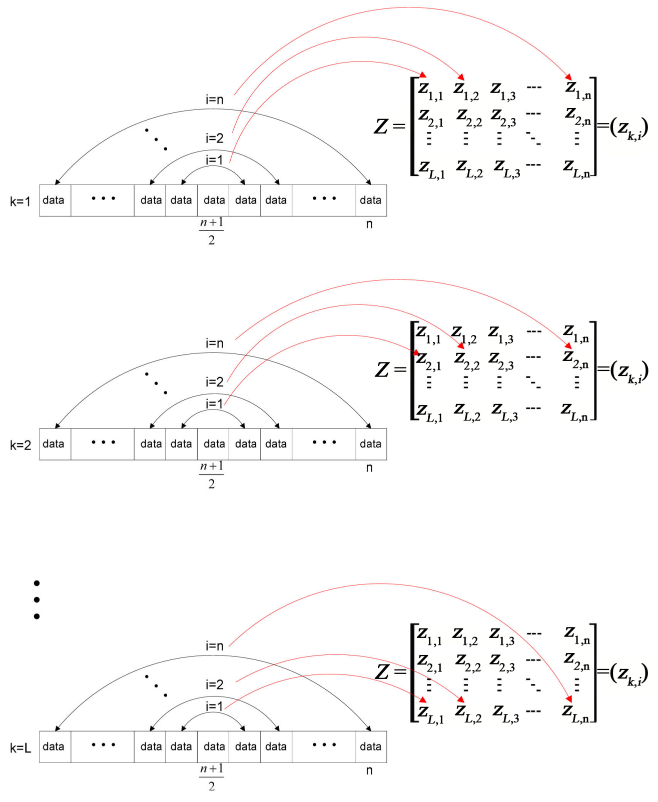

k=1
k=2
k=L
i=n
i=2
i=1
data
n
$\frac{n+1}{2}$
$Z = \begin{bmatrix} z_{1,1} & z_{1,2} & z_{1,3} & \cdots & z_{1,n} \\ z_{2,1} & z_{2,2} & z_{2,3} & \cdots & z_{2,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ z_{L,1} & z_{L,2} & z_{L,3} & \cdots & z_{L,n} \end{bmatrix} = (z_{k,i})$

Figure 2. Compare mechanism.

2.1.2. Positive Edge and Negative Edge Peak Points Detection

The target of this study is to detect peak points by applying the variance formula. After completing the formation of the Z matrix, the zero points are detected by applying the variance formula to each column. In some cases this value may not be zero due to noises; however, detection is done when the value is smaller than a minimal threshold value. Thus, if sigma (σ), in Equation (4) is found to be zero, then this point is interpreted to correspond to the peak value. If sigma is different from zero, this point does not correspond to the peak value,

$$\sigma_i = \frac{1}{\lambda - 1}\sum_{k=1}^{\lambda}\left[\left(z_{k,i} - \frac{1}{\lambda}\sum_{k=1}^{\lambda} z_{k,i}\right)^2\right]^{\frac{1}{2}} \tag{4}$$

where, for $i \in \{1,2,3,\cdots,n\}$ then, lambda in Equation (4) should be denoted as L so that all zero points of Z matrix given in Equation (4) are detected.

3. Implementation

3.1. Overview of the System

In the original AMPD algorithm, the best scale is automatically found by using four calculation steps Local Maxima Scalogram (LMS) calculation, Row-Wise summation of the LMS, LMS rescaling, and peak detection. On the other hand, it is necessary to reduce the amount of memory used in this AMPD algorithm while applying it to the FPGA in order to make a simple and effective pipeline design. The reason is that using more steps in application will increase memory usage and low latency. Original and modified algorithm flowcharts were given in Figure 1 and Figure 4 respectively. It was also mentioned that original algorithm flowchart in Figure 1 is more complex and has more steps than modified algorithm flowchart in Figure 4. In modified algorithm, step size is reduced and therefore the memory size is also reduced. If the time complexity (O) of algorithm is analyzed in terms of the flowchart given in Figure 3, total time complexity is found as $O(2n^2)$.

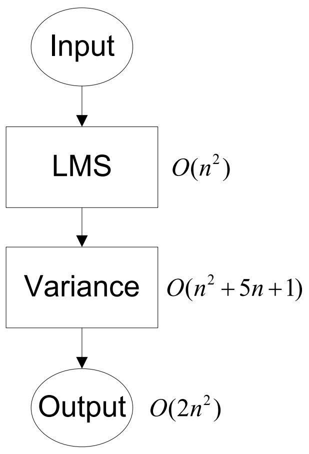
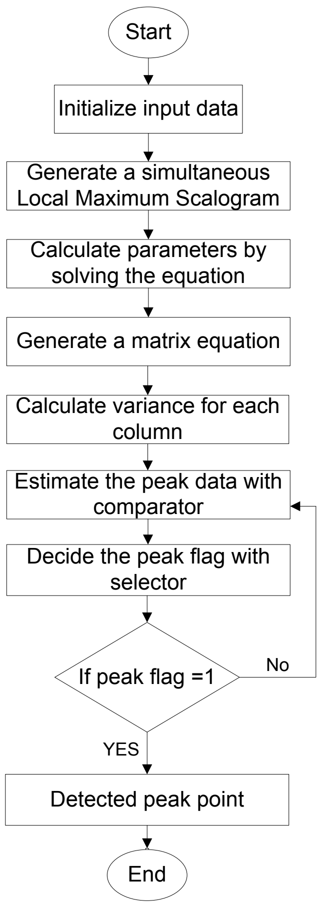


Figure 3. Time complexity of off-line algorithm flowchart.

Since the main target of this study is to apply AMPD algorithm to FPGA, a new algorithm has been proposed where the number of steps of AMD algorithm have been reduced. **Figure 4** shows the flowchart about the off-line algorithm. Therefore, there are only two steps in the proposed algorithms: Local Maxima Scalogram and peak detection for the best off-line calculations of peak points. The proposed system cannot straightforwardly calculate the peak points automatically with the low memory of FPGA. When the AMPD algorithm is applied on an FPGA, it can be reprogrammed for desired applications so that a logical gate is needed to make a process. Nevertheless, applying the deviation formula generates the LMS matrix and then all values are kept in registers. **Figure 5** shows the hardware design of the overview of implementation where k is the window scale.

In this design, a different element of the matrix in every clock cycle is compared and generated. Matrix generators with input X and output Z are serially connected to take advantage of the pipelining. After completion of generating all values, a basic peak flag is used to determine if a value corresponds to a zero point, that is, if a peak point is detected. Matrix generators are used to shift data sequentially, with the data for each matrix generator being compared with newly sampled data. In this manner, matrix elements of each scale are generated. In



Start
Initialize input data
Generate a simultaneous Local Maximum Scalogram
Calculate parameters by solving the equation
Generate a matrix equation
Calculate variance for each column
Estimate the peak data with comparator
Decide the peak flag with selector
If peak flag =1
No
YES
Detected peak point
End

Figure 4. Off-line algorithm flowchart.

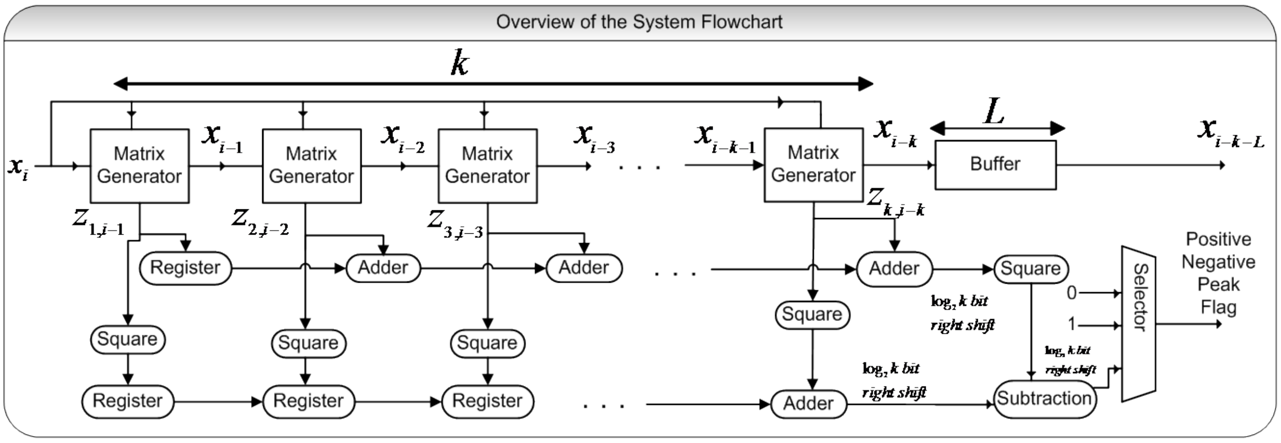


Figure 5. Overview of the algorithm flowchart.

one clock cycle, an element of the matrix will be generated, and as a result the summation and square summation are calculated sequentially. Finally, a division is performed to obtain an average and a squared average.

In the original AMPD algorithm, the standard deviation formula is utilized for detecting peak points. This involves rather complex arithmetic such as square root. To improve performance and efficiency of the FPGA implementation, the process is modified to use variance instead of standard deviation. The formula used in this design is in Equation (5), which shows the matrix designed by applying the variance formula to each column. Although both Equation (4) and Equation (5) are suitable for applying the variance, Equation (5) is used in this study due to its easy representation in hardware design.

$$\sigma_i = \frac{1}{L}\left[\left(\sum_{k=1}^{L} z_{k,i}^2\right) - \left(\frac{1}{L}\sum_{k=1}^{L} z_{k,i}\right)^2\right] \tag{5}$$

3.2. Matrix Generator and Decision Mechanism

Figure 6 depicts a Matrix Generator block diagram that is a decision mechanism to generate a matrix by using LMS calculation. It is also a critical part of the LMS calculation. The Matrix Generator requires a decision part for comparing values, which constitutes the major design of the matrix generator module. This module generates a matrix of one scale. The data generated by matrix generator for comparison reason is reduced to store and fed into the shift. The length of the shift register depends on the shift register's scale.

The input data is stored into the register and the data it is to be compared with will be inputted into the comparator in every clock cycle. Figure 7 is the expanded detailed hardware of the selector section of Figure 6 and it is the decision giving section on the achievement of positive and negative peak detections.

4. Results and Discussion

In this section, the original AMPD algorithm and the modified algorithm are evaluated in detail by comparing their detections of peak points of the same data

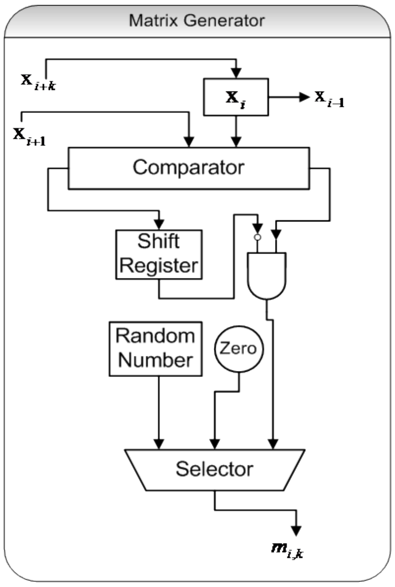


Figure 6. Matrix Generator system design.

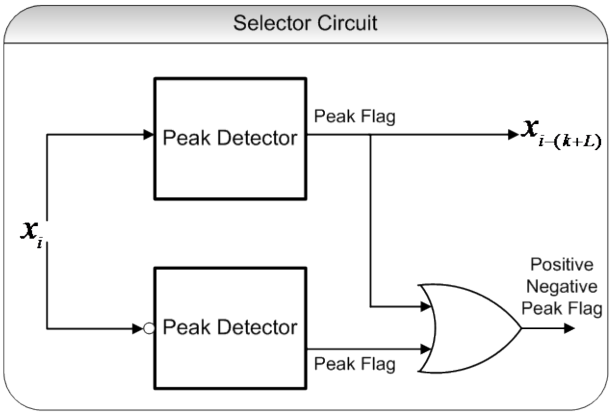


Figure 7. Selector circuit.

series. First of all, simulation results of both algorithms are given and then compared with each other. Later on, both original and modified algorithms are evaluated at different scales. After that, the sensitivity equation below is applied to each design to find their peak sensitivities.

$$\text{Sensitivity} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \tag{6}$$

where;

TP = True Positive.

FN = False Negative.

4.1. Simulation of the Original AMPD Algorithm

In this section, the peak points obtained from the original AMPD method as on-line have been introduced. A simulation has been performed with input data of the phase-to-phase effective voltage values of a medium-voltage transformer located in the Organized Industrial Zone used for the period from October 1 to October 31, 2015. The corresponding dataset contains 4470 data points recorded at 10-min intervals for each $L3$-$L2$, $L2$-$L1$ and $L1$-$L3$ phase-to-phase effective voltages. $L1$, $L2$ and $L3$ denote the power line in order in a 3-phase power system. The daily maximum and minimum peak points detected by the original AMPD method are shown in Figures 8(a)-(c) for $L3$-$L2$ ($V_{L3\text{-}L2}$), $L2$-$L1$ ($V_{L2\text{-}L1}$) and $L1$-$L3$ ($V_{L1\text{-}L3}$) line voltage values, respectively. When these figures are compared in detail, the original AMPD method detects the daily maximum peak points with the sensitivities of 93.75% for $V_{L3\text{-}L2}$, 96.96% for $V_{L2\text{-}L1}$ and 90.90% for $V_{L1\text{-}L3}$. Thus, high numbers of the daily maximum peak points are observed at the time of 06:20, 06:30, 06:50, 07:00, 07:10 and 07:50. Moreover, it detects the daily minimum peak points with the sensitivities of 93.75% for $V_{L3\text{-}L2}$, 96.77% for $V_{L2\text{-}L1}$ and 93.75% for $V_{L1\text{-}L3}$. In addition, identification of most of the daily minimum peak points are done at the time of 10:40, 11:00, 11:10, 17:30 and 19:00. Nevertheless, it should be noted that the original AMPD method introduced in this section has been designed in C Programming Language and it has been run as on-line. Max-Peak Sensitivities and Min-Peak Sensitivities concerning the line voltages are shown in Table 1.

4.2. Simulation of the Modified Off-Line Algorithm

In this section, a similar simulation as in 4.1 has been repeated for the designed Verilog algorithm using the same data. Hardware Description Language (HDL) simulations were performed with a Cadence NC-Verilog simulator. Figures 9(a)-(c) and Figures 10(a)-(c) illustrate the real data in red color with plus sign and modified Verilog algorithm data in green color with a star sign. However, in this case simulation results are obtained from the Verilog design algorithm as off-line.

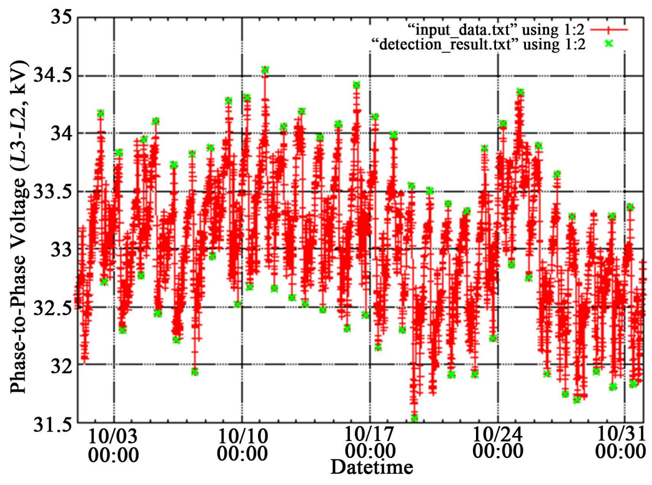
The daily maximum and minimum peak points detected by the modified AMPD method are depicted in Figures 9(a)-(c) for scale 33 for $L3$-$L2$ ($V_{L3\text{-}L2}$), $L2$-$L1$ ($V_{L2\text{-}L1}$) and $L1$-$L3$ ($V_{L1\text{-}L3}$) line voltage values, respectively.

The daily maximum and minimum peak points detected by the modified AMPD method are depicted in Figures 10(a)-(c) for scale 1025 for $L3$-$L2$ ($V_{L3\text{-}L2}$), $L2$-$L1$ ($V_{L2\text{-}L1}$) and $L1$-$L3$ ($V_{L1\text{-}L3}$) line voltage values, respectively.

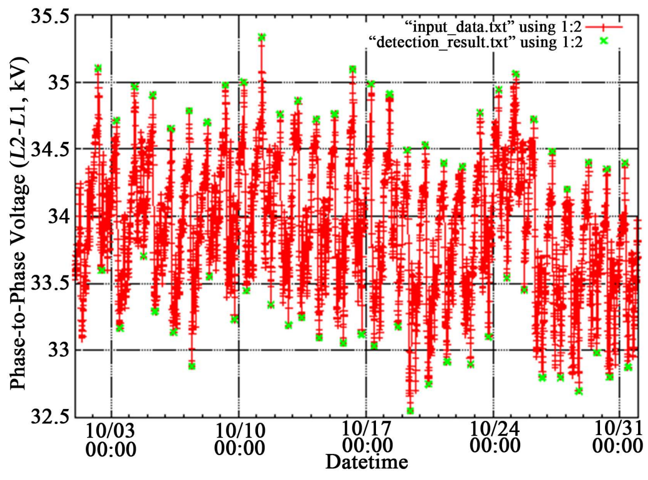
Table 2 represents the positive edge sensitivities at different scales. When the scale was increased, the sensitivities of $L3$-$L2$ line voltage were reduced gradually, but positive edge sensitivity of 87.87% was obtained at scales 33, 65 and 129 for $L2$-$L1$ line voltage values. Furthermore, the sensitivities of $L2$-$L1$ line voltage

Table 1. Max-peak sensitivities and min-peak sensitivities concerning the line voltages.

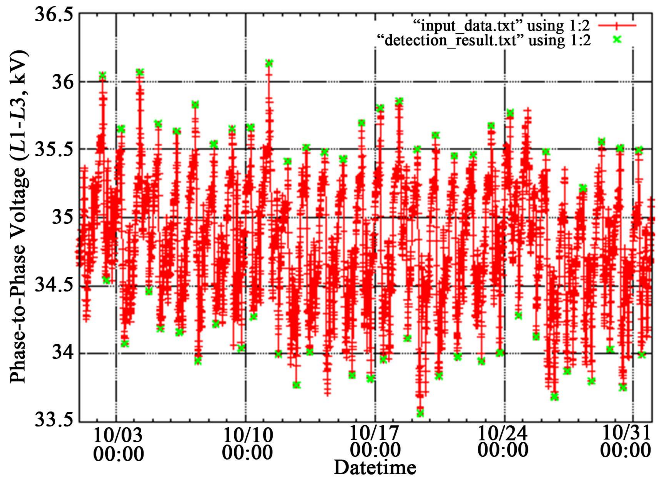
Phase-to-Phase Effective Voltages	Max-Peak Sensitivities	Min-Peak Sensitivities
$L3$-$L2$ line voltage values	93.75%	93.75%
$L2$-$L1$ line voltage values	96.96%	96.77%
$L1$-$L3$ line voltage values	90.90%	93.75%



(a)

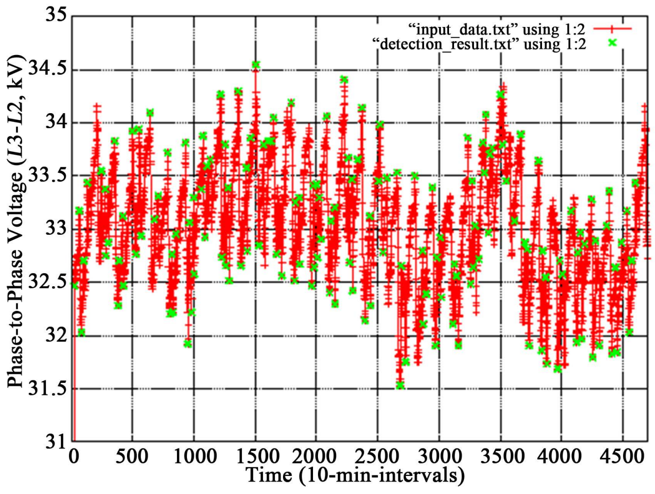


(b)

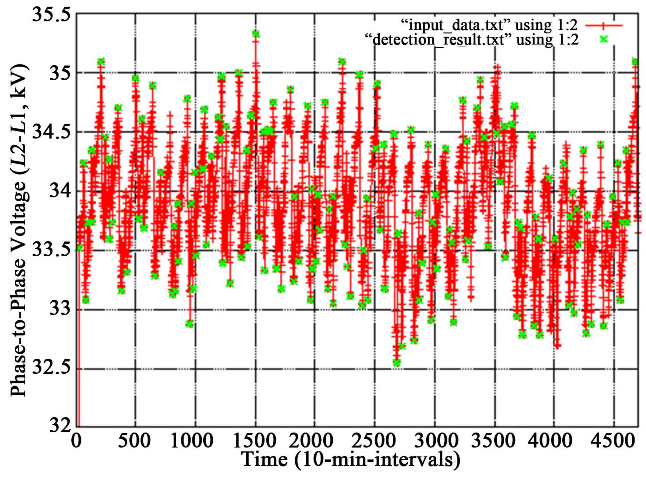


(c)

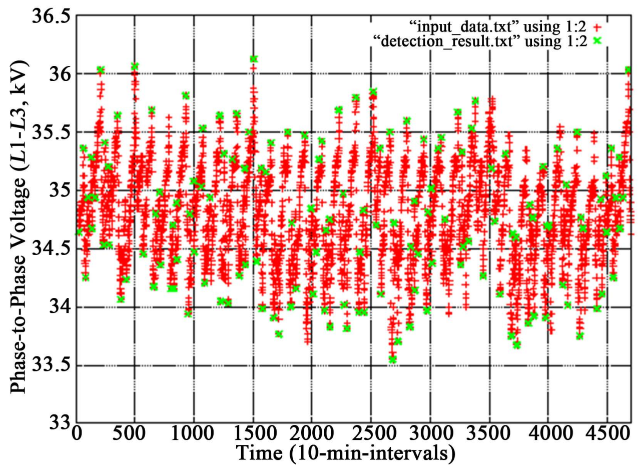
Figure 8. (a) Daily maximum and minimum peak points detected by the original AMPD method for *L*3-*L*2 line voltage values; (b) daily maximum and minimum peak points detected by the original AMPD method for *L*2-*L*1 line voltage values; (c) daily maximum and minimum peak points detected by the original AMPD method for *L*1-*L*3 line voltage values.



(a)

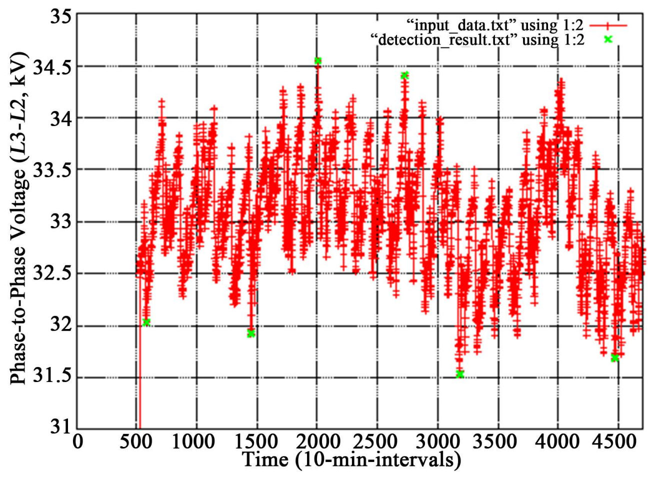


(b)

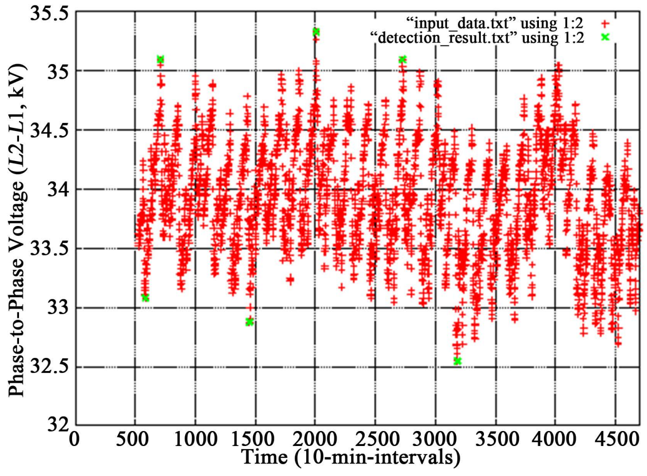


(c)

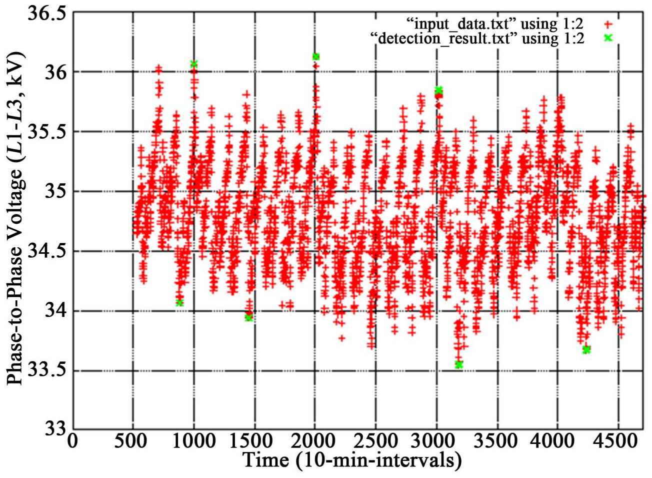
Figure 9. (a) Daily maximum and minimum peak points detected by the modified AMPD method for *L*3-*L*2 line voltage values; (b) daily maximum and minimum peak points detected by the modified AMPD method for *L*2-*L*1 line voltage values; (c) daily maximum and minimum peak points detected by the modified AMPD method for *L*1-*L*3 line voltage values.



(a)



(b)



(c)

Figure 10. (a). Daily maximum and minimum peak points detected by the modified AMPD method for *L*3-*L*2 line voltage values; (b) daily maximum and minimum peak points detected by the modified AMPD method for *L*2-*L*1 line voltage values; (c) daily maximum and minimum peak points detected by the modified AMPD method for *L*1-*L*3 line voltage values.

Table 2. Positive edge sensitivities for different scales and line voltages.

Sensitivities (Positive edge)	*L*3-*L*2 Line Voltage Values	*L*2-*L*1 Line Voltage Values	*L*1-*L*3 Line Voltage Values
Scale 33	84.37%	87.87%	81.81%
Scale 65	81.25%	87.87%	81.81%
Scale 129	78.125%	87.87%	81.81%
Scale 257	78.125%	75.75%	75.75%
Scale 513	12.5%	21.21%	21.21%
Scale 1025	6.25%	9.09%	9.09%

and *L*1-*L*3 line voltage remain constant for scales 33, 65 and 129. Nevertheless, they are also reduced at scales 257, 513 and 1025 because of the low frequencies of compared data at these scales. Finally, it is seen that maximum sensitivities are obtained at scales 33 and 65 for all line voltage values. They give almost the same results as the original algorithm.

Table 3 shows the negative edge sensitivities at different scales. As the scale is increased from 33 to 1025, the big changes in sensitivities have been observed for *L*1-*L*3 line voltage values. On the other hand, it was seen that maximum sensitivities of 93.54% were obtained at scales 33 and 65 for *L*2-*L*1 line voltage values that were almost the same as the original algorithm results. The sensitivities of *L*2-*L*1 line voltage values remain constant for scales 33 and 65. However, they are also reduced for the scales 129, 257, 513 and 1025 due to the low frequencies of compared data at these scales. Nevertheless, the sensitivities of *L*3-*L*2 line voltage values are constant for the scales 33, 65, 129 and 257, but they are also reduced for the scales 513 and 1025. Lastly, it is seen that negative edge sensitivities are obtained at scales 33 and 65 for all line voltage values. Almost the same results were obtained at scales 33 and 65 as the original algorithm.

Finally, sensitivities of all line voltages of *L*3-*L*2, *L*2-*L*1 and *L*1-*L*3 at higher scales were found very low due to very high sampling periods so that it caused missing the detection of peaks. This can also be explained simply by looking at Equation (1) and Figure 2, where it was seen that, when the scale is increased, the number of detected points are reduced due to an increased number of *L* in *Z* matrix. In another case, the sensitivity is also decreased at higher scales due to availability of noises in the signal.

4.3. Evaluation Environments and Method

In this section, the aforementioned hardware designed in Verilog HDL is explained and then device utilization and performance of the modified algorithm on the Kintex-7 XC7K325T [24] FPGAs are evaluated. As a mapping tool, a Vivado 2016.3 tool was used. Furthermore, the analog-to-digital converters (ADCs) transform analog electrical signals, generally the voltage amplitude, into a sequence of discrete values for data processing purposes. In this study, it was preferred to use a DC919af ADC with 100 MHz maximum system frequency

Table 3. Negative edge sensitivities for different scales and line voltages.

Sensitivities (Negative edge)	*L*3-*L*2 Line Voltage Values	*L*2-*L*1 Line Voltage Values	*L*1-*L*3 Line Voltage Values
Scale 33	75%	93.54%	87.5%
Scale 65	75%	93.54%	84.37%
Scale 129	75%	90.32%	78.125%
Scale 257	75%	67.74%	43.75%
Scale 513	21.87%	22.58%	25%
Scale 1025	12.5%	9.6%	12.5%

(sampling rate) and it was implemented on the FPGA board. The main target in this study was to increase the scale to observe and analyze the latency, memory usage and performance of the FPGA board. Therefore, different window lengths at various scales were designed to implement and analyze them on the FPGA board. Firstly, the bit size factor was fixed at 12 because there are many ADC compatible with 12 bits. Then the scale of input data was varied such as 33, 65, 129, 257, 513 and 1025 scales for comparison of resource usage, the result of which is shown in Table 4.

As seen in Table 4, when the scale is increased, an increase is detected directly from some slice logic utilization such as the number of slice LUTs, BRAMs, FFs, and DSP48E1 blocks, as well as the latency. In addition, design algorithm uses on-chip memory blocks (BRAMs) since the entire process is mapped on a pipelined structure based on shift registers. In terms of performance, the latency of peak detection was 23 clock cycles for the scale of 33. When each input data element has 12 bits, the maximum clock frequency is 144.927 MHz in Table 4. Furthermore, the maximum frequency was adjusted to 126.438 MHz in Table 4 by using scale 1025 input sources.

4.4. Evaluation of the AMPD Method with an FPGA Board

An evaluation of the AMPD method with an FPGA board is done in this section. Table 5 gives information on the performance of latency at 100 MHz timing constraint for two different scales. To state the purpose of detecting the peak points efficiently, the approach in this study achieves the real-time peak detection based on AMPD algorithm on an FPGA. When implementing AMPD algorithm on the FPGA board, bit size was chosen as 12, which was compatible with the ADC and 100 MHz maximum system frequencies.

Figure 11 shows the overview of the experiment system. First, FPGA sends the starting signal to the ADC. Then ADC sends 12 bits input data and 100 MHz system clock signal to the FPGA. Finally, the designed algorithms successfully detect all peak points as illustrated in Figure 14 and Figure 15. In particular, two different AMPD algorithms were designed with scales 65 and 33 due to having best sensitivities. Although the system can successfully detect peak points at both scales, the latency is taken into account for the performance. When scale

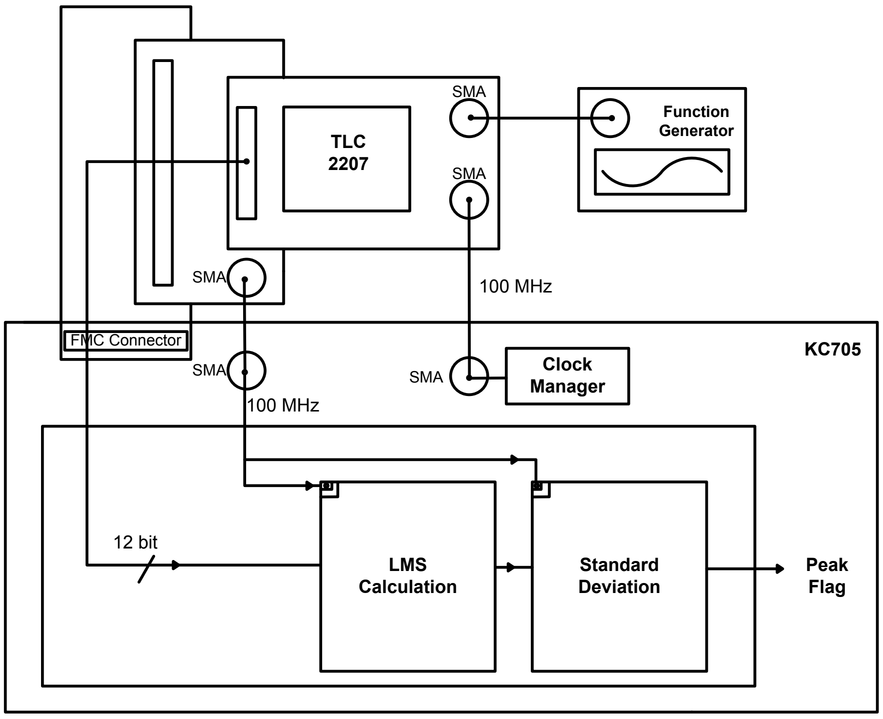


Figure 11. Overview of the proposed system.

Table 4. Resource usage with different scales.

Scale	Number of Slice LUTs	Number of BRAMs	Number of Flip-Flop	Number of DSP48E1	Max Frequency (MHz)	Latency (clock cycle)
33	2151	52	2910	22	144.927	23
65	3148	52	4370	32	145.50	40
129	5143	52	7306	64	156.66	73
257	9141	52	13,182	128	147.57	138
513	15,625	52	24,938	256	125.54	267
1025	31,467	52	48,454	512	126.438	524
Available	203,800	445	407,600	840	-	-

Table 5. Performance of different scales.

Scale	Bit Size	Upper Limit Frequency	System Clock Frequency	Latency Measure from Figure 9 and Figure 10
33	12	6 MHz	100 MHz	320 ns
65	12	3 MHz	100 MHz	502 ns

33 is selected, the latency is around 320 ns. This latency is a combination of ADC latency and algorithm calculation latency. When scale 65 is selected, total latency becomes around 502 ns because the window scale increases. Accordingly, the more the scale increases, the less the upper limit frequency becomes. The upper limit frequency is obtained by increasing the frequency from signal generator up

to losing output signal.

Table 5 provides information on latency and maximum upper limit frequency for different scales with a fixed bit size and system clock frequency. This table shows the evaluation result of the two different scales of 33 and 65. Furthermore, upper limit frequencies indicated the maximum time step of execution of the algorithm.

While the system is operating, the delay time is composed of two critical parts as algorithm calculation and converter as well as transmitting wire. For instance, when we implemented scale 65, we detected 502 ns total latency time from the oscilloscope screen depicted in Figure 14. To calculate its component for the calculation of algorithm, latency clock cycle at scale 65 in Table 4 is divided by constraint system clock frequency (100 MHz) first so that algorithm calculation time is found as 40/100 = 0.4 [μs] = 400 [ns]. Then, it was subtracted from the total latency time, 502 ns, read from the oscilloscope in Figure 14, in order to calculate the converter and transmitting wire part as 502 [ns] - 400 [ns] = 102 [ns]. Figure 12 shows detail about latency for scale 65.

This calculation is repeated for the scale 33 to make the distribution of latency time clearer and understandable. At scale 33, total latency time is measured as 320 ns from the oscilloscope screen in Figure 15. From Table 4, algorithm calculation time is found by dividing the latency clock cycle by constraint system clock frequency (100 MHz). So it is calculated as 23/100 = 0.23 [μs] = 230 [ns]. After that, when this time is subtracted from total latency measure, converter and transmitting wire part can be calculated as 320 [ns] - 230 [ns] = 90 [ns]. Figure 13 shows detail about latency for scale 33.

Figure 14 and Figure 15 are snapshots of experimental results with different scales on FPGA by using an oscilloscope. These results prove that a designed

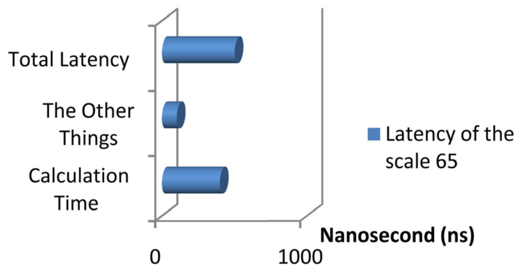


Figure 12. Latency of scale 65.

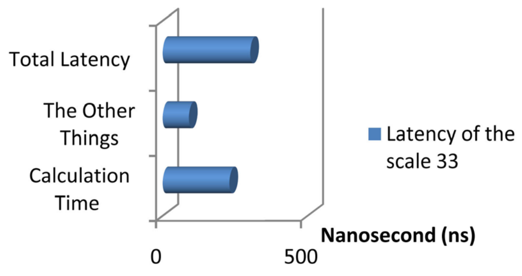
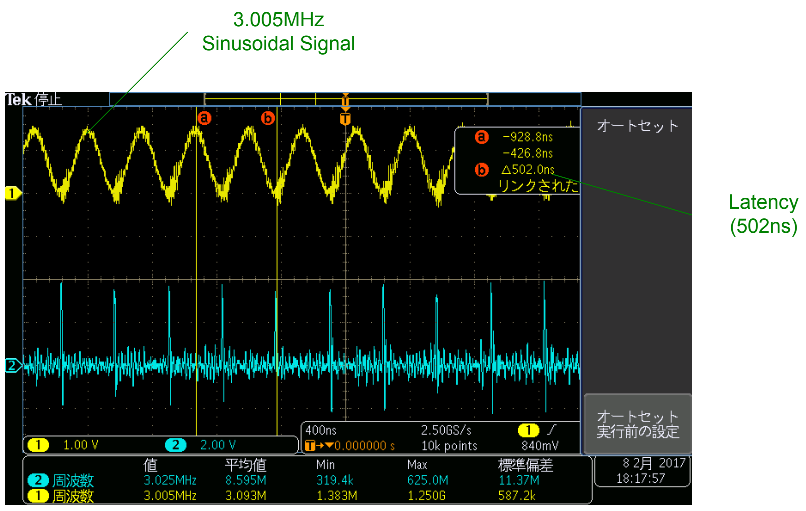
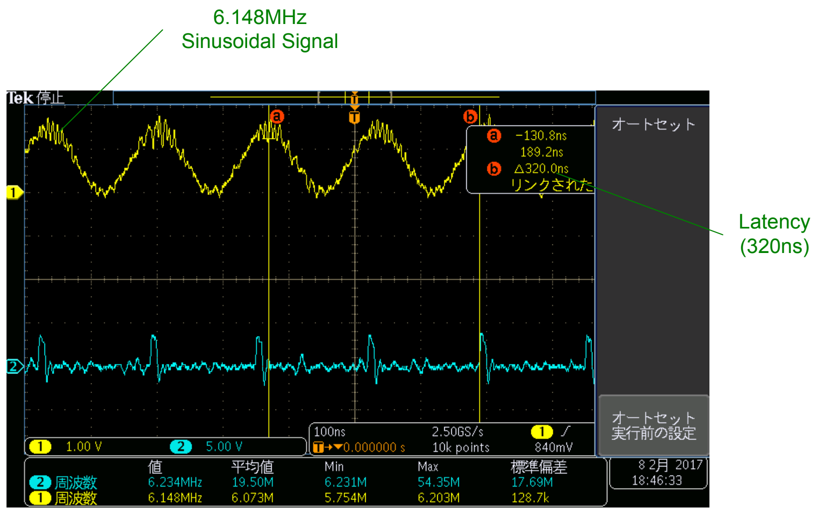


Figure 13. Latency of scale 33.



	値	平均値	Min	Max	標準偏差
2 周波数	3.025MHz	8.595M	319.4k	625.0M	11.37M
1 周波数	3.005MHz	3.093M	1.383M	1.250G	587.2k

Figure 14. Peak detect with 3.005 MHz sinusoidal signal.



	値	平均値	Min	Max	標準偏差
2 周波数	6.234MHz	19.50M	6.231M	54.35M	17.69M
1 周波数	6.148MHz	6.073M	5.754M	6.203M	128.7k

Figure 15. Peak detect with 6.148 MHz sinusoidal signal.

algorithm can effectively detect all peak points on the positive edge. **Figure 14** illustrates that designed algorithm can fully detect all peak points on the positive edge. This algorithm produces the peak points with synchronized 3.00 MHz sinusoidal signals. This implementation result shows scale 65, 12 bit size and around 502 ns total latency time and a maximum frequency of around 3 MHz. **Figure 15** shows all of the peak points on positive edge with synchronizing 6.148 MHz sinusoidal signal. This implementation result shows scale 33, 12 bit size, around the 320 ns latency time and a maximum frequency of around 6 MHz.

5. Conclusions

In this paper, a novel modified AMPD method was implemented on an FPGA. It was highlighted that the modified AMPD mechanism could be implemented as a pipelined hardware on an FPGA, and that fast detection latencies (320 ns and 502 ns for scales 33 and 65) could be achieved with a reasonable amount of hardware resources by slightly modifying the original on-line algorithm to fit the off-line processing. It was also demonstrated that the modified AMPD mechanism detects peak points from noisy time series data of the phase-to-phase effective voltage values of a medium-voltage transformer located in the Organized Industrial Zone.

Thus, modified AMPD algorithm has been proposed and evaluated by using different scales on an FPGA board. The proposed approach has achieved real-time peak detection based on AMPD algorithm on an FPGA. The latency was also presented at different scales from 33 to 1025. The codes written in real time were implemented and results were compared with simulation results to achieve the main target of this study. Although both scales can effectively detect the peak points, the latency was taken into account for the performance on the ADC and algorithm calculating time. So peak sensitivities obtained from on-line original AMPD were compared with the sensitivities obtained from off-line modified algorithm using the same data. It was observed that scales 33, 65, 129 and 257 produced more or less successful results. In terms of the peak sensitivities, scales 33 and 65 produced similar results as the original AMPD that was proposed in this study. That means the proposed novel mechanism can be used as an off-line robust and real-time peak detection algorithm by combining the AMPD algorithm and the FPGA technology.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Scholkmann, F., Boss, J. and Wolf, M. (2012) An Efficient Algorithm for Automatic Peak Detection in Noisy periodic and Quasi-Periodic Signals. *Algorithms* 2012, **5**, 588-603. https://doi.org/10.3390/a5040588

[2] Jacobson, M. (2001) Auto-Threshold Peak Detection in Physiological Signals. Technical Report, DTIC Document.

[3] Ding, F., Booth, C.D. and Roscoe, A.J. (2016) Peak-Ratio Analysis Method for Enhancement of LOM Protection Using M-Class PMUs. *IEEE Transactions on Smart Grid*, **7**, 291-299. https://doi.org/10.1109/TSG.2015.2439512

[4] Belkaid, A., Gaubert, J.P. and Gherbi, A. (2017) Design and Implementation of a High-Performance Technique for Tracking PV Peak Power. *IET Renewable Power Generation*, **11**, 92-99. https://doi.org/10.1049/iet-rpg.2016.0023

[5] Du, P., Kibbe, W.A. and Lin, S. (2006) Improved Peak Detection in Mass Spectrum by Incorporating Continuous Wavelet Transform-Based Pattern Matching. *Bioinformatics*, **22**, 2059-2065. https://doi.org/10.1093/bioinformatics/btl355

[6] Nenadic, Z. and Burdick, J.W. (2005) Spike Detection Using the Continuous Wavelet Transform. *IEEE Transactions on Biomedical Engineering*, **52**, 74-87. https://doi.org/10.1109/TBME.2004.839800

[7] Goodfellow, J., Escalona, O.J., Kodoth, V., Manoharan, G. and Bosnjak, A. (2016) Denoising and Automated R-Peak Detection in the ECG Using Discrete Wavelet

Transform. *Computing in Cardiology Conference* (*CinC*), Vancouver, 1045-1048.

[8] Das, S., Mukherjee, S., Chatterjee, S. and Chatterjee, H.K. (2016) Noise Elimination and ECG R Peak Detection Using Wavelet Transform. *Ubiquitous Computing, Electronics & Mobile Communication Conference* (*UEMCON*), IEEE Annual, New York, 1-5. https://doi.org/10.1109/UEMCON.2016.7777876

[9] Thiamchoo, N. and Phukpattaranont, P. (2016) Application of Wavelet Transform and Shannon Energy on R Peak Detection Algorithm. 13*th International Conference on Computer, Telecommunications and Information Technology* (*ECTI-CON*), Chiang Mai, 1-5. https://doi.org/10.1109/ECTICon.2016.7561280

[10] Li, Q., Zhang, W., Li, M., Niu, J. and Wu, Q.M.J. (2017) Automatic Detection of Ship Targets Based on Wavelet Transform for HF Surface Wavelet Radar. *IEEE Geoscience and Remote Sensing Letters*, **14**, 714-718. https://doi.org/10.1109/LGRS.2017.2673806

[11] Khan, G.K. and Sawant, A.G. (2016) Spartan 6 FPGA Implementation of 2D-Discrete Wavelet Transform in Verilog HDL. 2016 *IEEE International Conference on Advances in Electronics, Communication and Computer Technology* (*ICAECCT*), Pune, 139-143. https://doi.org/10.1109/ICAECCT.2016.7942570

[12] Palshikar, G. (2009) Simple Algorithms for Peak Detection in Time Series. *Proc.* 1*st Int. Conf. Advanced Data Analysis*, *Business Analytics and Intelligence*, Ahmedabad.

[13] Khatri, K.L. and Tamil, L. (2017) Early Detection of Peak Demand Days of Chronic Respiratory Diseases Emergency Department Visits Using Artificial Neural Networks. *IEEE Journal of Biomedical and Health Informatics*, **22**, 285-290. https://doi.org/10.1109/JBHI.2017.2698418

[14] Sumukha, B.N., Kumar, R.C., Bharadwai, S.S. and George, K. (2016) A Novel Approach to Peak Detection Using Sequential Learning Algorithm. 2*nd International Conference on Contemporary Computing and Informatics* (*IC*3*I*), IEEE, Noida, 1-6.

[15] Crema, C., Depari, A., Flammini, A. and Vezzoli, A. (2016) Efficient R-Peak Detection Algorithm for Real-Time Analysis of ECG in Portable Devices. *Sensors Applications Symposium* (*SAS*), IEEE, Catania, 1-6.

[16] Dora, C. and Biswal, P.K. (2016) Robust ECG Artifact Removal from EEG Using Continuous Wavelet Transformation and Linear Regression. 2016 *International Conference on Signal Processing and Communications* (*SPCOM*), IEEE, Bangalore, 1-5. https://doi.org/10.1109/SPCOM.2016.7746620

[17] Annam, J.R. and Surampudi, B.R. (2016) Inter-Patient Heart-Beat Classification Using Complete ECG Beat Time Series by Alignment of R-Peaks Using SVM and Decision Rule. *International Conference on Signal and Information Processing* (*IConSIP*), IEEE, Vishnupuri, 1-5. https://doi.org/10.1109/ICONSIP.2016.7857480

[18] Aqil, M., Jbari, A. and Bourouhou, A. (2016) Adaptive ECG Wavelet Analysis for R-Peaks Detection. 2016 *International Conference on Electrical and Information Technologies* (*ICEIT*), IEEE, Tangiers, 164-167. https://doi.org/10.1109/EITech.2016.7519582

[19] Park, J.S., Lee, S.W. and Park, U. (2017) R Peak Detection Method Using Wavelet Transform and Modified Shannon Energy Envelope. *Journal of Healthcare Engineering*, **2017**, 1-14. https://doi.org/10.1155/2017/4901017

[20] Qin, Q., Li, J., Yue, Y. and Liu, C. (2017) An Adaptive and Time-Efficient ECG R-Peak Detection Algorithm. *Journal of Healthcare Engineering*, **2017**, 1-14. https://doi.org/10.1155/2017/5980541

[21] Singh, A., Dubey, S. and Bhatia, M. (2013) Design and Simulation of FPGA Based Digital System For Peak Detection and Counting. *IJARCSSE*, **3**, 804-807.

[22] Ma, M. (2005) Developing and Implementing Phase Normalization and Peak Detection for Real-Time Image Registration. Ph.D. Thesis, Master's Thesis, Delft University of Technology.

[23] Colak, A.M., Shibata, Y. and Kurokawa, F. (2016) FPGA Implementation of the Automatic Multiscale Based Peak Detection for Real-Time Signal Analysis on Renewable Energy Systems. 5*th International Conference on Renewable Energy Research and Applications* (*ICRERA*), Birmingham, 379-384.

[24] Kintex-7 XC7K325T Evaluation Kit.
https://www.xilinx.com/products/boards-and-kits.html