Bulletin of Electrical Engineering and Informatics
Vol. 11, No. 2, April 2022, pp. 613~623
ISSN: 2302-9285, DOI: 10.11591/eei.v11i2.3293

Design of FFNN architecture for power quality analysis and its complexity challenges on FPGA

Prathibha Ekanthaiah[1], Mesfin Megra[1], Aswathnarayana Manjunatha[2], Likhitha Ramalingappa[2]
[1]Department of Electrical Power and Control Engineering, Adama Science and Technology University, Adama, Ethiopia
[2]Department of Electrical and Electronics Engineering, Sri Krishna Institute of Technology, Bengaluru, India

Article Info

Article history:

Received Nov 7, 2021
Revised Jan 22, 2022
Accepted Mar 8, 2022

Keywords:

Artificial neural network
Computation complexity
Feed forward neural network
Field programmable gate array
Power quality

ABSTRACT

As we all know, power quality (PQ) issues are a major concern these days. Field programmable gate array (FPGA) are essential in PQ analysis, particularly in smart meters for data processing, storage, and transmission. One of the most significant advantages of FPGA is its reconfigurability, with vast hardware resources that can be used to implement complex as well as time-critical data processing units. Because the FPGA architecture supports fixed point arithmetic, data loss occurs in the data path unit, necessitating the realization of the PQ event detection module, and classification model to be more accurate than software implementation algorithms. The majority of the work reported, with feed forward neural network (FFNN) structure occupying large number of multipliers and adders for classification, most of the work reported has not addressed to minimize the data path resources for FFNN instead have addressed in improving classification accuracy. Based on these issues, this paper addresses the implementation challenges in FFNN architecture design by proposing improved and fast architectures. The proposed FFNN architecture design using optimum resources. The FFNN based classifier are designed to perform PQ event detection and classification with 99.5% accuracy. The FFNN processor operates at maximum frequency of 238 MHz.

Corresponding Author:

Prathibha Ekanthaiah
Department of Electrical Power and Control Engineering, Adama Science and Technology University
Adama, Oromia, Ethiopia
Email: prathibha.astu@gmail.com

1. INTRODUCTION

As we know that utilities, end users, manufacturers, and all other consumers are concerned about electric power quality (PQ). Economic losses are primarily caused by poor electricity quality. According to the survey results. Voltage swell, voltage sag, harmonics, and transients are the most typical power quality disturbance events in power systems. Industrial equipment, household equipment, and other natural failures are the main causes of power quality issues. Because of the network's interconnection, these disturbances will spread like wildfire [1]–[3]. In this regard, monitoring systems can be useful at every stage of the power system [4]. As a result, it was necessary to determine when the PQ disturbance occurred as well as characterize the disturbances with wavelet and other approach [5]–[9]. We require monitors to detect and classify PQ occurrences, which can be accomplished by smart revenue meters, in order to pinpoint the root source of problems. Smart meters [10] in a smart grid system are used to monitor and detect power quality issues. The major job activity of a smart meter is the automatic identification of disturbances and the reporting of events [11]–[13]. So, for these PQ event detections, wavelets are crucial, and neural networks are commonly utilized for categorization. An artificial neural network (ANN) classifier was used to classify power quality problems [14], [15]. Under this technique, different types of power quality disturbances were investigated, as well as flawless PQ disturbances being classified using a trained neural network that requires fewer training samples [16]–[18]. The development of field programmable gate array (FPGA) that incorporate higher order statistics (HOS) processing cores to provide a signal analysis intended to detect and as well as an as an ANN to classify the power quality disturbances (PQD) in addition to standard energy tariff calculations and FPGA based method for training feed forward neural networks (FFNN) [19], [20].

Smart meters with inbuilt display unit assist in determining the electricity consumption in every house hold providing information for management of energy resources. Unwarranted disturbances in power line causes damages to smart meters that need to be addressed with suitable security solutions. Physically unclonable function (PUF) is a new concept in smart meters that provides flexibility in reconfiguration of hardware models for security reasons [21]. Most of the new generation smart meters are based on PUF concepts as they support reconfigurability [22]. Neural networks have been extensively used for forecasting of local disaggregated loads in buildings, micro grids, and distribution areas [23]. Neural network based power quality analysis to classify events such as sag, swell and interruptions is presented considering real time data obtained from 132 kV bus [24]. There are very few literatures on FPGA implementation of neural network architecture for smart meter applications. PQ events have been investigated using wavelet-based methods. One of the novel methods uses complex wavelets for PQ signal analysis. In this method the wavelet sub bands of both real and imaginary coefficients are considered for feature extraction [25]. The features in the wavelet sub band are distinct for PQ events such as harmonics and flicker, for events such as sag and swell, sag with flicker, or swell with flicker it is required to use neural network approaches for classification of near distinct features. In 2017, Prathibha *et al*. [26] features identified from complex wavelet sub bands for PQ events ae classified using neural network approaches. The computation complexity of neural network-based approach limits is use for real time applications. Because the FFNN structure requires a large number of multipliers and adders for classification, most of the work reported has focused on improving classification accuracy rather than minimizing data path resources for FFNN. Based on the observations made above, this work addresses the implementation challenges in FFNN architecture design using the most efficient resources.

2. ARTIFICIAL NEURAL NETWORKS DESIGN FOR PQ EVENT CLASSIFICATION

2.1. Neural network classifier of power quality event

The PQ event detection and classifier algorithm is presented in Figure 1. The algorithm comprises of two stages, feature detector with dual-tree complex wavelet transform (DTCWT) and classifier using ANN. Power signal is first pre-processed and is decomposed to multiple sub bands using DTCWT filters bank structure. From the sub bands, appropriate features are extracted and are used to train the classifier. The trained classifier is designed to classify the PQ events.

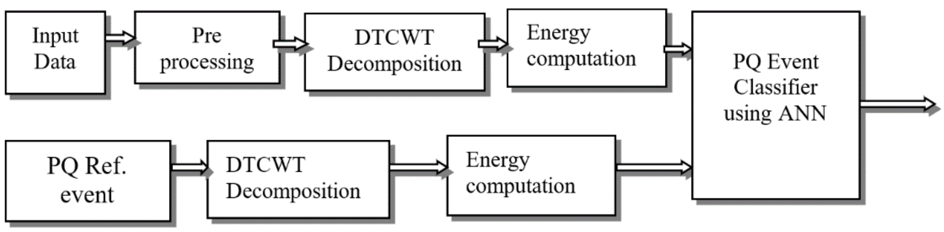

Figure 1. Block diagram of PQ event detector and classifier

In order to identify features that represent PQ events and correspondingly train the classifier it is required to generate PQ events using mathematical models. PQ events such as sag, swell, harmonics, interrupts, sag with harmonics, and swell with harmonics are generated using parametric equations that are considered as reference event. Each signal generated is grouped into frames each of 2048 samples. DTCWT processes the frames to compute 10 sub bands and from each of the sub band energy features are computed that represents the PQ event. The neural network-based classifier is trained based on these features and the optimum weights are computed. From the trained and designed neural network-based classifier PQ events are classified and characterized. Figure 2 presents the proposed methods for real time classification of PQ events using neural network approach. The neural network is initially trained to classify PQ events from the data base. By performing training, it is identified that the network is able to reach its global minima point and optimum weights and biases are identified for classification. The optimum weight and bias matrix obtained is recorded and stored for the trained network for the corresponding PQ event classification.

The real-time neural network (RNN) classifier comprises of NN classifier consisting 16 hidden layers and 4 output layers. Each of the networks designed and assigned with corresponding weights and biases obtained from the training process. The RNN classifier is trained to classify six PQ events generated based on parametric equations. The trained RNN is used for classification of real time PQ events. The multiplexer unit selects the PQ event to be classified. The RNN model is to be designed and implemented on hardware. There are several NN models that have been successfully used for classification of PQ events but are limited to software modelling. Hardware implementation of NN based classifiers is presented in this work. The NN classifier needs to be designed optimizing area, timing and power requirements. The primary objective of this work is design of NN based architecture.

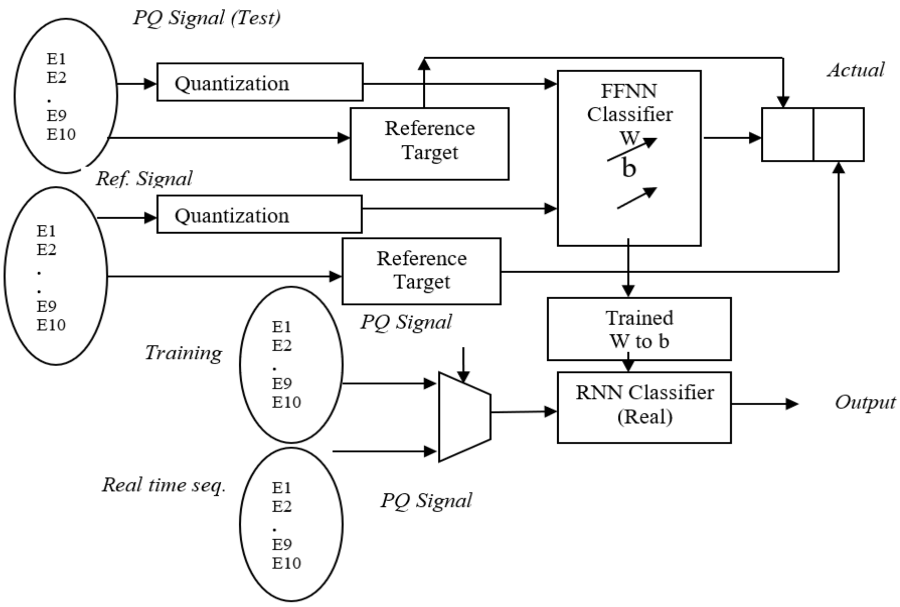

Figure 2. Neural network training method

2.2. Artificial neural network model

ANNs are selected for PQ event classification as they are found to be more robust once they are trained with large number of data sets. In addition to training data sets it is also required to design the neuron structure by selecting appropriate number of neurons and network transfer functions. In this paper FFNN architecture with 10 inputs, 16 neurons in the hidden layer, and 4 neurons in the output layer is designed for classification of PQ events. The network architecture is shown in Figure 3. The hidden layer outputs are denoted by $\{a_1, a_2, \ldots\ldots a_{16}\}$ and the corresponding weights and biases are represented by $W_{n,m}$ and b_n respectively where n represents the neuron and m represents input. The hidden layer neuron output is represented as HE_n. $HE_n=f(a_n)$ and a_n is represented by (1),

$$a_1=E_1W_{1,1}+E_2W_{1,2}+\ldots+E_{10}W_{1,10}+b^1{}_1 \quad (1)$$

The hidden layer network function is tan sigmoid function. In general, the intermediate outputs a_k are represented by (2),

$$a_k=\sum_{i=1}^{10} (E_iw_{i,k})+ b^1{}_{k,i},\ k=1,2,3,4\ldots\ldots16 \quad (2)$$

Similarly, the output layer output is mathematically represented by (3) and the network function is purelin.

$$O_k=\sum_{i=1}^{16} (HE_iw_{k,i})+ b^2{}_{k,i}\ ,\ k=1,2,3,4 \quad (3)$$

Figure 3 presents the proposed architecture which consists of six FANN architectures. The level-1 FANN output will be either target for undistorted PQ signal or target for sag. If the network in level-1 generates any other output other than these the output is discarded. The level-2 network is trained to classify the inputs to corresponding PQ events. Thus, the two-level training improves reliability in classification process. The level-1 network is defined as coarse classifier and is responsible for classification of some six PQ events like sag, swell, harmonics, interrupts, and the fine classifier at level-2 is responsible for improving the accuracy in classification as the inputs to this classifier consists of 6 inputs each one selected from the output of coarse classifier. As there are six different FFNN modules that are trained to classify six events by generating four outputs, the first stage neural network architecture is called as coarse classifier.

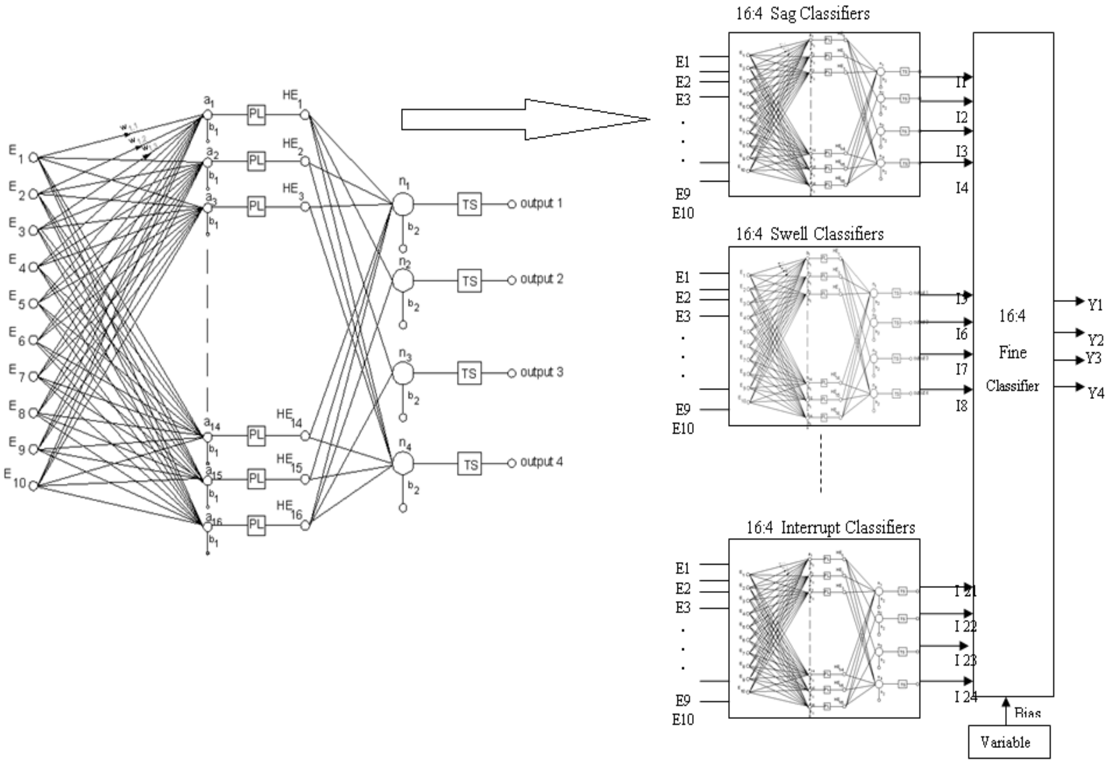

Figure 3. Feed forward network architecture for PQ events classification

3. COMPLEXITY CHALLENGE IN FFNN ARCHITECTURE

With two stage classifier (coarse and fine) and coarse classifier consisting of six FFNN structures with each structure consisting of 10 inputs, 16 hidden layer neurons, and 4 output layer neurons the computation complexity of FFNN architecture is two-fold. The hidden layer will have 10 multipliers per neuron and 10 adders overall with 10 network functions. Total number of multipliers per FFNN structure will be 160 which are required to be stored in 160 registers. The output layer will have 16 inputs processed by four neurons that will require 16 multipliers per neurons and 64 total numbers of multipliers, 4 adders and four network functions. The total number of multipliers per FFNN is estimated to be 224, and every FFNN requires 14 adders and 14 network functions. The propagation delay of every FFNN is estimated to be 6T. Where T denotes the delay time, the hidden layer will have delay of 3T and the output layer will also have delay time of 3T. As there are three stages of data processing modules (multiplier, adder, and network function) the processing delay of each stage is T time period. The coarse classifier with six FFNN modules processes the data into 24 output samples that are further processed by the fine classifier. The fine classifier consists of 24 inputs, M neurons in the hidden layer, and four neurons in output layer. The number of multipliers required are 24*M + M*4, adders are M+4 and network functions are also M+4. With three stages of processing the total delay of fine classifier is 3T. The output of coarse classifier is processed by the fine classifier and hence the total propagation delay of the FFNN classifier is 9T. The computation complexity of FFNN classifier is dependent on selection of number of neurons in the hidden layer and selection of network functions. During training phase, it is required to evaluate performances of various network structures based on selection of neurons and network functions. In this paper, FFNN architectures are designed that can operate at high frequencies and reduced computation complexity. Design of efficient FFNN architectures is presented in next section.

4. RESEARCH METHOD

4.1. Design of feed forward neural network architecture

The FFNN architecture shown in Figure 4 is a two-stage multi-layered structure. The first stage is made of six FFNN multi-layered structure each of the FFNN comprises of 10 input layer samples, 16 hidden layers neurons, and 4 output layer neurons. The number of weights and biases required for hidden layer are 160 and 16 respectively and for output layer are 64 and 4 respectively. The number of multipliers (M) and Adders (A) for one FFNN in stage one i.e., coarse classifier is 224 and 20 respectively. Considering all six FFNNs the multipliers and adders required are 6M and 6A. Similarly for second stage FFNN structure i.e., the fine classifier, there are 24 inputs, 64 hidden layer neurons, and 4 output layer neurons. The number of multipliers and adders required are 1792 and 68 respectively. Implementation of two stage classifier on hardware will occupy area and also increase power dissipation. In order to optimize power dissipation and reduce computation complexity novel architecture is proposed.

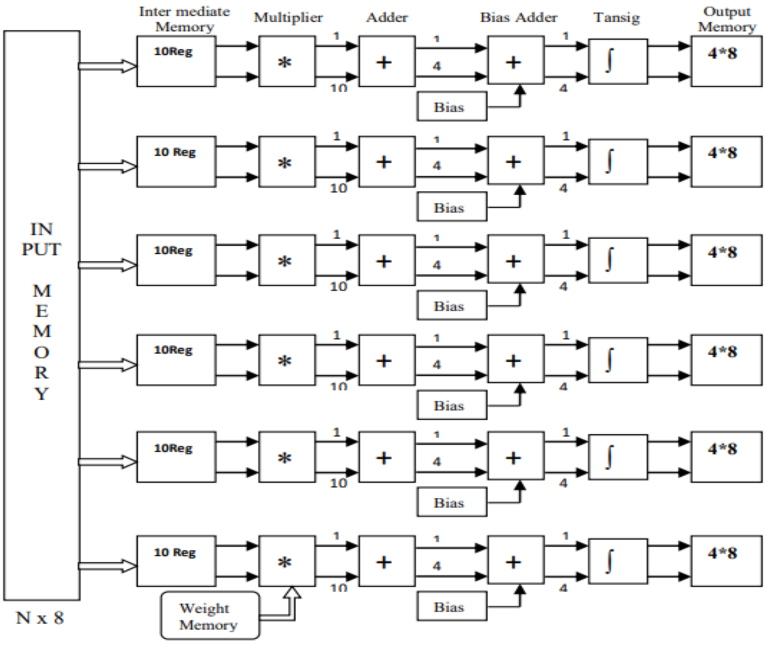

Figure 4. Hidden layer FFNN architecture with parallel processing

4.2. Design of optimal structure for coarse classifier

The direct architecture for first stage FFNN is shown in Figure 4. The energy levels computed by considering DTCWT coefficients are stored in the input memory. The first set of 10 energy levels is read into six group of intermediate memory array, with each 135-group comprising of 10 registers. As there are six FFNNs for six different PQ events are discussed for the coarse classifier, each of the FFNN will have hidden layer and output layer. The 10 energy levels will be processed by the hidden layer and the output layer to generate four outputs per FFNN. The hidden layer of every FFNN processes data from 10 register contents in the multiplier array with each multiplier array consisting of 160 multipliers. The corresponding weights are stored in weight memory array registers. The multiplication operation generates 160 partial products that are required to be accumulated by adder array consisting of 159 adders. The output of adder array is processed by the network function to generate the final output. The four outputs from every FFNN hidden layer are further processed by the output layer that consists of 16 multipliers every neuron. The 16 partial products are accumulated by the adder array consisting of 15 adders and the output of adder array is processed by the network function. The FFNN generates four outputs for every 10 inputs fed at the input layer. In addition to multiplier operation at the multiplier array stage, these data are further added with bias elements. Each of the six FFNN structure will have 10 registers in the input layer, the hidden layer is shown in Figure 4 will consist of 160 multipliers in the multiplier array, 159 adders in the adder array, 16 adders for biases, 16 network functions and the output layer are shown in Figure 5 will consist of 64 multipliers in the multiplier array, 63 adders in the adder array, 4 adders for biases, and 4 network functions. With each of the inputs in the input register being represented by 8-bit signed representation, fixed point arithmetic is used for FFNN architecture design.

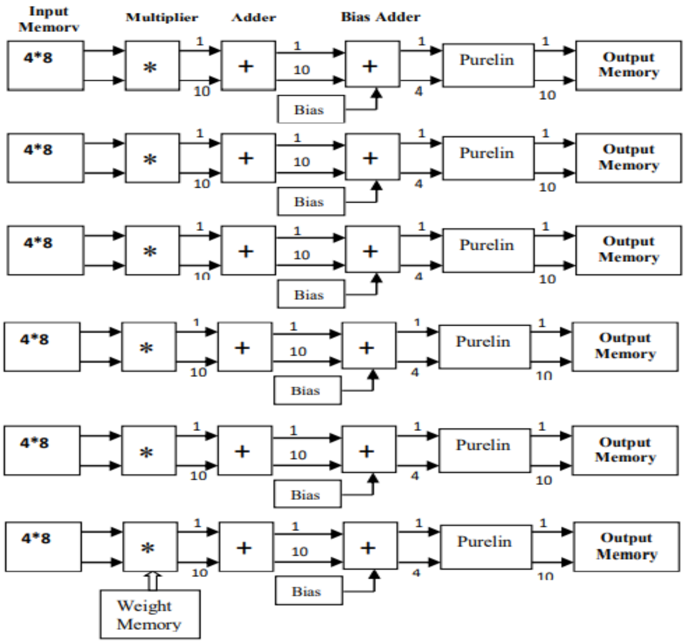

Figure 5. Output layer FFNN architecture with parallel processing

4.3. Design of optimal hidden layer structure

In order to reduce the large number of arithmetic units required for design of every FFNN structure, a novel method is proposed and implemented in this work. The proposed hidden layer structure is presented in Figure 6. The proposed hidden layer structure consists of 10 multipliers that are designed to operate in parallel multiplying two operands one from the input data and the other the weight element. The 10-multiplier array performs multiplication of 10 input elements with corresponding 10 weight elements and the 10 partial products are accumulated in the adder array which has four stages of addition operation. The 10 product terms are accumulated in the first stage adder that has five adders to generate five outputs. The five outputs are accumulated in the second stage adder (has two adders) to generate three outputs and the third stage adder array (has one adder) generates two outputs and the final stage adder array (has one adder) generates the final output. The bias operations also need to be carried out and hence the bias operation is performed along with the accumulation of partial products in the third stage by inserting an additional adder as shown in Figure 6.

The weight storage register array consists of 160 locations that are grouped into 10 register array elements represented as RAs. Each RAs store the 10 weight elements corresponding to the input data of every neuron. The bias array is of depth 16 and stores 16 bias elements in the bias array structure denoted as B1 to B16. During the first 10 clock cycles 10 weights are loaded into the weight register array denoted by W1-W10 from the RA1. The multiplier array performs multiplication of weights with the 10-input data from the input array register during 11^{th} clock cycle. As there are four stages of addition, accumulation operation of multiplied products is carried out in four clock cycles (12^{th} clock to 15^{th} clock), during which the first bias element from the bias register array is loaded into the bias register. At the end of 16^{th} clock the network function processes the data and generates the final output from the look up table (LUT) to store the results in the output register array. The LUT is designed to perform the task of tansig function, predetermined outputs of tansig function for all possible inputs in the range -5 to +5 are stored in the LUT, which is of depth 256. The output register is of depth 16 and is a register array structure.

The modified structure shown in Figure7 is designed to compute sixteen outputs of hidden layer using a single neuron structure. As the input remains constant for all the 16 neurons, the weight elements and bias elements are correspondingly loaded into the weight register. For the first neuron to compute the output 10 weights and one bias element that is corresponding to first neuron is loaded into the modified neuron architecture. The hidden layer structure generates the first output and is stored in the output register. Similarly for the second neuron output computation, the 10 elements in the RA2 are loaded into the weight registers (W1-W10) and BA2 is loaded into bias register. After 16 iterations, the modified neuron structure computes the 16 outputs of hidden layer structure that are stored in the output register. For computation of every hidden layer output, input data is first loaded along with weights, and bias that is completed in 10 clock cycles. The multiplier array stage requires one clock cycle, adder array requires four clock cycles, and network function requires one clock cycle. In total, for computation of one output after data is loaded into the unit, it requires 16 clock cycles. After first output is computed, 10 clock cycles are required for loading the weights of each neuron. To complete computation of 16 neuron outputs total clock cycles required are 160 adders the modified structure is realized. In the modified structure four neurons are realized using single neuron structure as shown in Figure 6. The register array consists of 64 registers grouped into four groups of 16 each. The bias elements for the hidden layer are of 4 and are stored in the bias register of depth 4.

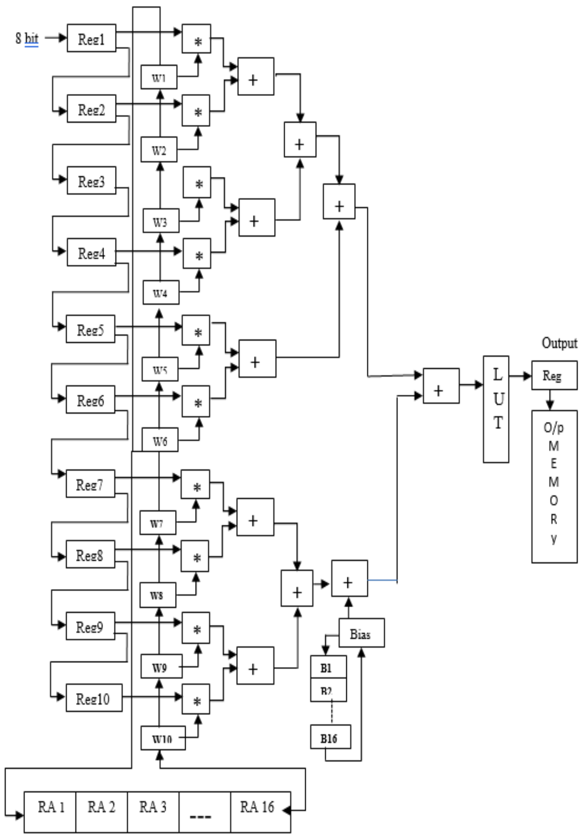

Figure 6. Modified hidden layer structures

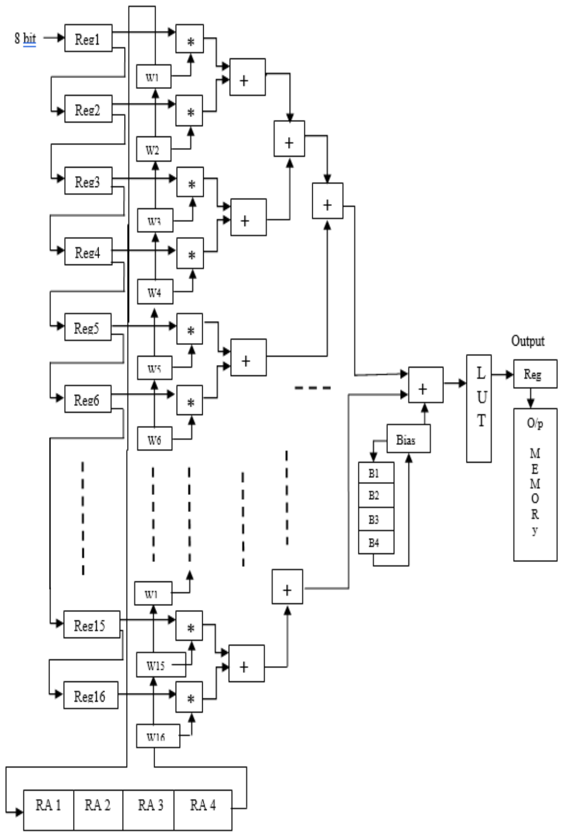

Figure 7. Pipeline architecture of single neuron

With 16 weight elements from the RA register is loaded into the weight registers the multiplier array performs multiplication of input data with the corresponding weight elements to generate 16 products. The adder array comprising of 5 stages of adders computes the accumulation of 16 products using 8 adders in first stage, 4 adders in second stage, 2 adders in third stage, 1 adder in the fourth stage, and finally the bias is added in the fifth stage. The accumulated output is processed by the network function to generate the final output of FFNN. The number of clock cycles to generate one output of four outputs is estimated to be of 23 clock cycles (16 clocks for weight element loading, 1 clock for multiplication, five clocks for addition, and one clock for network function). After 92 clocks the output layer generates the four outputs for the FFNN structure. Table 1 compares the arithmetic complexity of proposed FFNN architecture with direct form implementation.

Table 1. Comparison of FFNN structure

Parameters	Direct form–hidden layer	Proposed–hidden layer	Direct form–output layer	Propose–output layer
Input registers	10	10	16	16
Weight registers	160	160	64	64
Bias registers	16	16	4	4
Multipliers	160	16	64	16
Adders	160	16	64	16
Network functions	16	16	4	4
Total number of clock cycles	16	160	23	92

The proposed modified FFNN architecture design reduces the number of multipliers and adders for the hidden layer by 90% with latency of 160 clock cycles and the output layer multiplier and adder is reduced by 75% with latency of 92 clock cycles. As there are six FFNN structures in the coarse classifier the total optimization in terms of multiplier and adder is compared in Table 2. From the comparisons presented in Table 2 the number of adders and multipliers for the coarse classifier is reduced by 85.71%. The number of clock cycles required for computation i.e., the latency of the network is increased by 84%. The proposed architecture is modeled using Verilog and is simulated using Xilinx integrated synthesis environment (ISE). Known input vectors are fed into the test bench and the outputs obtained are compared with the theoretical values. The logic correctness is verified manually and the code is further synthesized for FPGA implementation.

Table 2. Comparison of coarse classifier structure

Parameters	FFNN–coarse classifier	
	Direct form	Modified structure
Multipliers	1344	192
Adders	1344	192
Network functions	120	120
Total number of clock cycles	39	252

5. RESULTS AND DISCUSSION

The section showcases the simulation results of the hardware model for the proposed FFNN classifier. The simulations are performed in ISE simulator. The functionality of hardware models is cross verified with that of results obtained from Chip Scope Pro.

The weights and biases obtained after training the network is represented using fixed point number represented and the input is represented in 2's complete signed representation. The Verilog coding is carried out for both the coarse and fine classifier and is verified for its functionality by comparing the results with the results obtained in MATLAB environment. The simulation results of design using ISE sim are as shown in Figure 8. The functionally verified design is implemented on Virtex-5 FPGA and the intermediate results are captured from the hardware environment using Chip Scope debugging tool. The Xilinx Chip Scope Pro tools are added to the Verilog design to capture input and output directly from the FPGA hardware. The Chip Scope simulations are as shown in Figure 9 is compared with the simulation results of ISE Sim and is validated. The design is again synthesized using Xilinx ISE and implemented on Virtex-5 FPGA board. Figure 10 presents the synthesized net list of coarse classifier structure that is implemented on Virtex-5 FPGA. Similarly, the fine classifier is also implemented on Xilinx FPGA after validation using Chip scope results. Table 3 presents the synthesis results of proposed FFNN architecture synthesized using Xilinx ISE targeting Vertex 5 FPGA consisting of 110 million gates. The proposed design occupies less than 18% of the FPGA resources and consumes a total power of 0.44 W, operating at a maximum frequency of 238 MHz (4.201 ns).

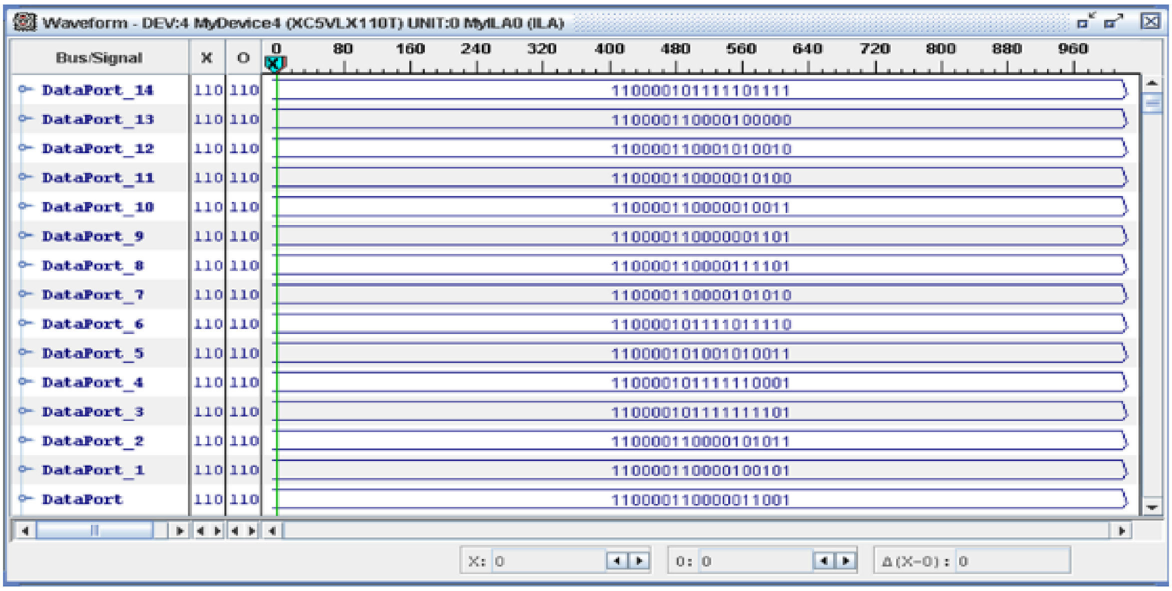

Figure 8. Simulation results of ISE Sim

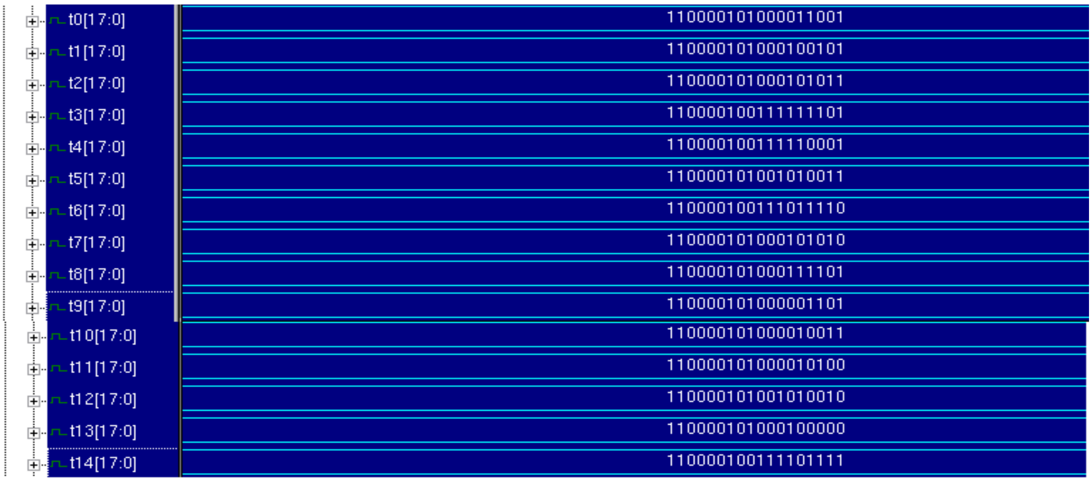

Figure 9. Hidden layer simulation in ChipScope Pro

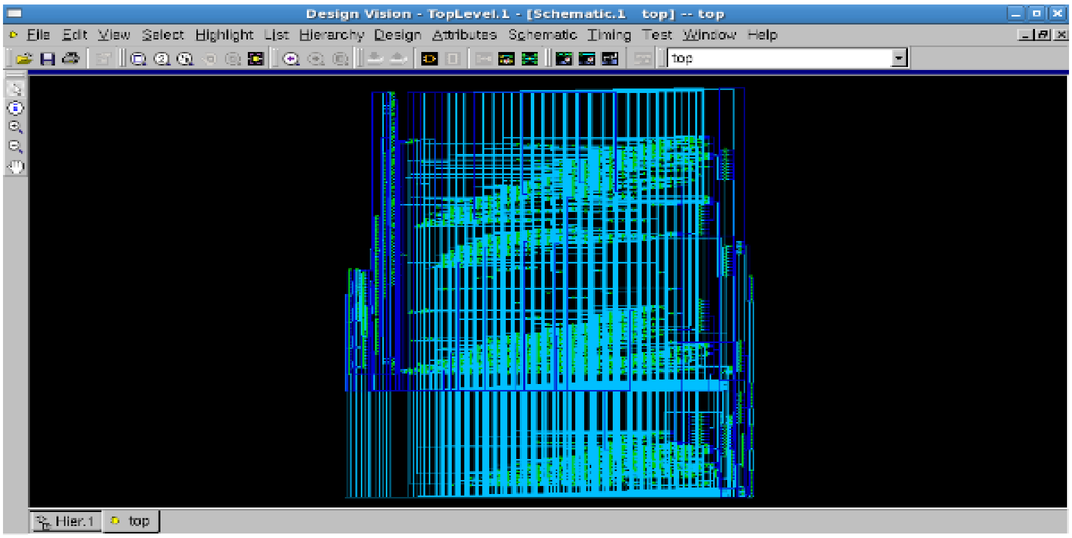

Figure 10. Synthesized net list of FFNN

Table 3. FFNN FPGA synthesis results

Selected device	5VLX110TFF1136-1		
Slice logic utilization	Number of slice LUT's	12444 out of 69120	18%
	Number used as logic	12444 out of 69120	18%
I/O utilization	Number of I/Os	192	
	Number of bonded IOBs	192 out of 640	30%
	IOB flip flops/latches	15	
	Number of BUFG/BUFGCTRLS	6 out of 32	18%
Timing report	Min I/P arrival time before clock	5.341 ns	
	Max O/P reqd. time after clock	1.024 ns	
	Max combinational path delay	7.230 ns	
Power report	Total quiescent power	0.53905 W	
	Total dynamic power	0.01380W	
	Total power	0.55285W	

6. CONCLUSION

Smart systems for smart grid that can monitor power quality by measuring PQ events occurring are required to detect the event and classify the events. The FFNN based classifier is designed to perform PQ event detection and classification with 99.5% accuracy. The FFNN processor operates at maximum frequency of 238 MHz the total resource utilization of FFNN is less than 23% of the total resources available.

In addition to these modules, it is also required to design energy computation module, thresholder, and quantization logic. Memory unit for storage of input data and intermediate data is also required to be considered. The present paper addresses the hardware implementation of FFNN cores on FPGA platform. Interfacing FFNN with all other glue logic modules will required first-in, first-out (FIFO) architecture and data synchronization network. The proposed designs for FFNN can be used as intellectual property (iP) cores for any signal and control applications.

REFERENCES

[1] S. Chattopadhyay, M. Mitra, and S. Sengupta, “Electric Power Quality (Power Systems) 2011th Edition,” *Publisher: Springer*, 2011, pp. 1-177.
[2] P. Caramia, G. Carpinelli, and P. Verde, “*Power quality indices in liberalized markets*,” John Wiley & Sons, 2009.
[3] D. K. Dubey, “Issues and challenges in electricity sector in India,” *International Conference on Issues in Emerging Economies (ICIEE)*, vol. 5, no. 4, pp. 132–139, January 2015.
[4] S. H. Laskar, “Power Quality Monitoring by Software Development,” *International Conference on Power System Engineering (ICPSE2010)*, Penang, Malaysia, Feb. 2010.
[5] O. Poisson, P. Rioual, and M. Meunier, "Detection and measurement of power quality disturbances using wavelet transform," in *IEEE Transactions on Power Delivery*, vol. 15, no. 3, pp. 1039–1044, July 2000, doi: 10.1109/61.871372.
[6] T. K. Abdel-Galil, M. Kamel, A. M. Youssef, E. F. El-Saadany, and M. M. A. Salama, "Power quality disturbance classification using the inductive inference approach," in *IEEE Transactions on Power Delivery*, vol. 19, no. 4, pp. 1812–1818, Oct. 2004, doi: 10.1109/TPWRD.2003.822533.
[7] H. He and J. A. Starzyk, "A self-organizing learning array system for power quality classification based on wavelet transform," in *IEEE Transactions on Power Delivery*, vol. 21, no. 1, pp. 286–295, Jan. 2006, doi: 10.1109/TPWRD.2005.852392.
[8] M. Uyar, S. Yildirim and M. T. Gencoglu, “An effective wavelet-based feature extraction method for classification of power quality disturbance signals,” *Electric Power Systems Research*, vol. 78, no. 10, pp. 1747–1757, October 2008, doi: 10.1016/j.epsr.2008.03.002.
[9] M. Mishra, “Power quality disturbance detection and classification using signal processing and soft computing techniques: a comprehensive review,” *International Transactions on Electrical Energy Systems*, vol. 29, no. 8, p. e12008, March 2019, doi 10.1002/2050-7038.12008.
[10] S. H. Laskar, S. Khan, and Mohibullah, "Power quality monitoring in sustainable energy systems," *2012 IEEE International Symposium on Sustainable Systems and Technology (ISSST)*, 2012, pp. 1–6, doi: 10.1109/ISSST.2012.6228002.
[11] Q. Cetina, R. A. J. Roscoe, and P. S. Wright, "Challenges for Smart Electricity Meters due to Dynamic Power Quality Conditions of the Grid: A Review," *2017 IEEE International Workshop on Applied Measurements for Power Systems (AMPS)*, 2017, pp. 1–6, doi: 10.1109/AMPS.2017.8078345.
[12] M. Campbell, N. Watson, and A. Miller, “Smart Meters to Monitor Power Quality at Consumer Premises,’’ *EEA Conference & Exhibition*, 2015.
[13] F. A. S. Borges, R. A. S. Fernandes, I. N. Silva, and C. B. S. Silva, "Feature Extraction and Power Quality Disturbances Classification Using Smart Meters Signals," in *IEEE Transactions on Industrial Informatics*, vol. 12, no. 2, pp. 824–833, April 2016, doi: 10.1109/TII.2015.2486379.
[14] M. Ijaz, M. Shafiullah, and M. A. Abido, "Classification of power quality disturbances using Wavelet Transform and Optimized ANN," *2015 18th International Conference on Intelligent System Application to Power Systems (ISAP)*, 2015, pp. 1–6, doi: 10.1109/ISAP.2015.7325522.
[15] C. -M. Li, Z. -X. Li, N. Jia, Z. -L. Qi, and J. -H. Wu, "Classification of Power-Quality Disturbances Using Deep Belief Network," *2018 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, 2018, pp. 231–237, doi: 10.1109/ICWAPR.2018.8521311.
[16] Z. M. Chen, M. S. Li, T. Y. Ji, and Q. H. Wu, "Detection and classification of power quality disturbances in time domain using probabilistic neural network," *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 1277–1282, doi: 10.1109/IJCNN.2016.7727344.
[17] G. D. J. Martinez-Figueroa, D. Morinigo-Sotelo, A. L. Zorita-Lamadrid, L. Morales-Velazquez, and R. D. J. Romero-Troncoso, "FPGA-Based Smart Sensor for Detection and Classification of Power Quality Disturbances Using Higher Order Statistics," in *IEEE Access*, vol. 5, pp. 14259–14274, 2017, doi: 10.1109/ACCESS.2017.2732726.
[18] S. Jaiswal and M. S. Ballal, “FDST-based PQ event detection and energy metering implementation on FPGA-in-the-loop and NI-Lab VIEW,” *IET Science, Measurement & Technology*, vol. 11, no. 4, pp. 453–463, 2017, doi: 10.1049/iet-smt.2016.0481.
[19] S. N. A. Foumani, “An FPGA Accelerated Method for Training Feed-forward Neural Networks Using Alternating Direction Method of Multipliers and LSMR,” *arXiv preprint arXiv:2009.02784*, 2020.
[20] X. Wei, Y. Liang, and J. Cong, "Overcoming Data Transfer Bottlenecks in FPGA-based DNN Accelerators via Layer Conscious Memory Management," *2019 56th ACM/IEEE Design Automation Conference (DAC)*, 2019, pp. 1–6.
[21] M. Majzoobi, F. Koushanfar, and M. Potkonjak, “Techniques for design and implementation of secure reconfigurable PUFs,” in *ACM Trans. on Reconfigurable Technology and Systems*, vol. 2, no. 1, pp. 1–33, 2009, doi: 10.1145/1502781.1502786.
[22] M. M. Abutaleb and A. M. Allam, “Secure LowCost FPGA-based AMI System using LTE Technology*,” CiiT International Journal of Networking and Communication Engineering*, vol. 4, no. 7, pp. 377–383, June 2012.
[23] O. Valgaev, F. Kupzog, and H. Schmeck, “Adequacy of neural networks for wide-scale day-ahead load forecasts on buildings and distribution systems using smart meter data,” *Energy Informatics*, vol. 3, no. 1, pp. 1–17, 2020, doi: 10.1186/s42162-020-00132-6.
[24] P. Patil, K. Muley, and R. Agrawal, "Identification of Power Quality Disturbance Using Neural Network," *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2019, pp. 990–996, doi: 10.1109/ICECA.2019.8822214.
[25] E. Prathibha, A. Manjunatha, and C. P. Raj, “PQ Data Compression Algorithm with Modified Quantizer and Adaptive Band logic using DTCWT,” *Archives of Electrical Engineering*, vol. 67, no. 1, pp. 207–223, 2018, doi: 10.24425/119002.

[26] E. Prathibha, A. Manjunath, and C. P. Raj, "PQ event detection and classification based on dual tree complex wavelet and coarser-fine two stage ANN classifier," *Journal of Theoretical and Applied Information Technology*, vol. 95, no. 23, pp. 6624–6634, December 2017.

BIOGRAPHIES OF AUTHORS

Prathibha Ekanthaiah did her B.E during 2000 from Kuvempu University and M. Tech in VLSI Design & Embedded System during 2010 and Ph.D in 2018 in power quality from Viswesvaraya technological University, Karnataka. She is working in the Department of Electrical and Computer Engineering, Adama Science and Technology university, Adama, Ethiopia. Published 13 International and National journals, 28 International and National Conferences. Her Research areas of interest are power quality, smart grid, and renewable energy. She is the life member of Indian society for technical education. She can be contacted at email: prathibha.astu@gmail.com.

Mesfin Megra was born in Bishoftu, Ethiopia on January 7, 1989. He is currently a lecturer and the Department Head of Electrical Power and Control Engineering at Adama Science and Technology University in Adama. He previously worked as a lecturer in the Department of Electrical Engineering at Debre Brhan University in Debre Brihan (Ethiopia). Electrical Engineer at Fortune Engineering Plc in Addis Abeba (Ethiopia) and Distribution construction operation Electrical Engineer I at Ethiopian Electric Utility in Addis Abeba (Ethiopia). Many papers have been published in international journals and at conferences. He can be contacted at email: mesfinmegra@gmail.com.

Aswathnarayana Manjunatha received his BE in 1984 from Mysore University and his ME in Power Systems in 1992 from Bangalore University. Dr. MGR Deemed University Chennai awarded him a Ph.D. in power system in 2011. He is currently the principal of the Sri Krishna Institute of Technology in Bangalore, Karnataka, India. His research interests include power quality, power systems, and renewable energy. 30 international and national journals have been published, as have 45 international and national conferences. He is a life member of the Indian Society for Technical Education and an engineer member of the Institute of Engineers (India). He can be contacted at email: manjuprinci@gmail.com.

Likhitha Ramalingappa earned her BE in 2008 from PES College of Engineering in Mandya, which is part of Visvesvaraya Technological University. Visvesvaraya Technological University awarded me an M.Tech. in Power System in 2011. Visvesvaraya Technological University is where she is pursuing her Ph.D. in Power Quality Analysis. In 2011, she began her career as an Assistant Professor in the Department of Electrical and Electronics Engineering at Sri Krishna Institute of Technology in Bangalore. Her research interests include power quality, power systems, and distributed generation. 7 international and national journals have been published, as have 12 international and national conferences. She can be contacted at email: likhitha.pes@gmail.com.