Artificial Neural Network Architecture Design Using Vedic Multipliers

A.Mounica[1], E.Rama Krishna[2], Dr.R.Rama Chandra[3]
[2] HOD & Assistant Professor, Dept of ECE
[3] Principal & Professor
[1, 2, 3] Sri Krishna Devaraya Engineering College,Gooty,A.P

Abstract- *An Artificial neural network (ANN) is parallel Information processing structure consists of processing units. The processing unit decides while the network is efficient or not. So it is needed to design an efficient processing unit which provides better performance. The processing unit consists of MAC unit (Multiplication and Accumulation) and Activation unit. In an existing system, the processing MAC unit was designed by Booth multiplier and carry look ahead adder. The existing processing unit provides delay and consumes more area and power. To overcome the drawbacks, a new processing unit, with Vedic multiplier consisting of square root carry select adder (SQRT-CSLA) is designed. The proposed design overcomes the drawbacks of the existing system, and it also provides better performance to the entire network. The Activation function unit was designed by sigmoid neurons process. Entire processing unit was implemented and verified by using Verilog HDL language*

I. ARTIFICAL NEURAL NETWORKS (ANNs)

A. Neural Network Architecture

An Artificial Neural Network is a parallel information processing consists of processing units. The neural network was changed to Artificial Neural Networks, because it's not dealing with biological neural networks. ANN was deals with general computing architecture known as Multiple Instruction Multiple Data (MIMD) parallel processing architecture.

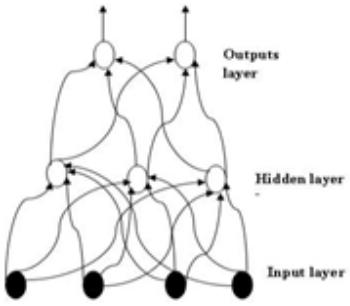

Fig. 1.1Simple neural network

Fig. 1.1shows the architecture of simple neural network, it contains three layers, input, hidden and output layer. Input data is feed forward to the output via the hidden layer. In-between the input and output, processing is performed by the help of processing unit.

B. Framework for ANN model

There are different ANN models but each model can be specified by the following aspects:

- A set of processing units
- A state of activation for each unit
- An output for each unit
- Topology of the network
- An activation rule to update the activities of each unit
- An external environment provides information to the network
- A learning rule to modify the structure of connectivity by using information provided by the external environment.

After the processing of information, the output function uses the activation value to calculate the output of the unit. A simple artificial neuron tells how the process is done in the processing MAC unit. MAC operation is important one to get an accurate results from the neural networks.

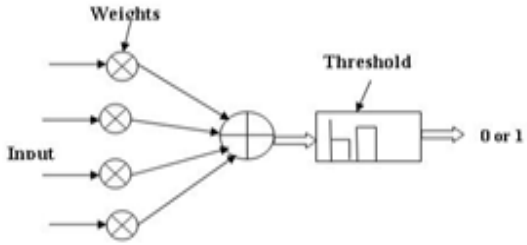

Fig. 1.2 Simple artificial neuron

Fig 1.2 shows the simple artificial neuron in the artificial neural network. It contains multiplication and accumulation unit. Each input are multiplied with weights individually, output from the multiplier is added by using

addition unit. Added output is forward to activation unit, based on the threshold it produced the output 0 or 1.

II. PROPOSED SYSTEM

The existing processing unit provides delay and consumes more area and power, to overcome this, a new processing unit is designed which contains Vedic multiplier with square root carry select adder (SQRT-CSLA). The proposed design overcomes the drawbacks of the existing system, and it also provides better performance to the entire network. The Activation function unit was designed by sigmoid neurons process. Entire processing unit was implemented and verified by using Verilog HDL language

.

2.2.1 Vedic Mathematics

Vedic mathematics is part of four Vedas (books of wisdom). It is part of Sthapatya- Veda (book on civil engineering and architecture), which is an upa-veda (supplement) of Atharva Veda.

It covers explanation of several modern mathematical terms including arithmetic, geometry (plane, co-ordinate), trigonometry, quadratic equations, factorization and even calculus. His Holiness JagadguruShankaracharyaBharati Krishna Teerthaji Maharaja (1884-1960) comprised all this work together and gave its mathematical explanation while discussing it for various applications. Swamiji constructed 16 sutras (formulae) and 16 Upa sutras (sub formulae) after extensive research in Atharva Veda. The very word "Veda" has the derivational meaning i.e. the fountainhead and illimitable storehouse of all knowledge. Vedic mathematics is the name given to the ancient system of mathematics or, to be precise a unique technique of calculations based on simple rules and principles with which many mathematical problems can be solved, be it arithmetic, algebra, geometry or trigonometry.

The system is based on 16 Vedic sutras or aphorisms, which are actually word formulae describing natural ways of solving a whole range of mathematical problems. The beauty of Vedic mathematics lies in the fact that it reduces the cumbersome-looking calculations in conventional mathematics to a very simple one. This is so because the Vedic formulae are claimed to be based on the natural principles on which the human mind works. This is a very interesting field and presents some effective algorithms which can be applied to various branches of engineering such as computing and digital signal processing.

Vedic Mathematics existed in ancient India and was rediscovered by a popular mathematician, Sri Bharati Krishna Tirthaji. He divided Vedic mathematics into 16 simple sutras (formulae). These Sutras deal with Arithmetic, Algebra, Geometry, Trigonometry, Analytical Geometry etc. The simplicity in the Vedic mathematics sutras paves way for its application in several prominent domains of engineering like Signal Processing, Control Engineering and VLSI.

i. Anurupye) Shunyamanyat -If one is in ratio, the other is zero.
ii. ChalanaKalanabyham -Differences and similarities.
iii. EkadhikinaPurvena- By one more than the previous One.
iv. EkanyunenaPurvena -By one less than the previous one.
v. Gunakasamuchyah-Factors of the sum is equal to the sum of factors.
vi. Gunitasamuchyah-The product of sum is equal to sum of the product.
vii. NikhilamNavatashcaramamDashatah -All from 9 and last from 10.
viii. ParaavartyaYojayet-Transpose and adjust.
ix. Puranapuranabyham -By the completion or noncompletion.
x. Sankalana- vyavakalanabhyam -By addition and by subtraction.

Shesanyankena Charamena- The remainders by the last digit implementation of 4X4 bit Vedic multiplier which uses the 2X2 bit Vedic multiplier as a basic building block.

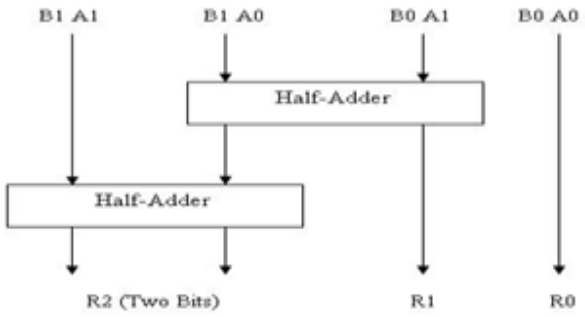

Fig. 2.8 2X2 Vedic Multiplier

The same method can be extended for input bits 4 and 8. But for higher number of bits in input, little modification is required. The structure of 16 bit Vedic multiplier by using 8X8 Vedic multiplier is illustrated in Fig 2.9. This structure consists of three 16 bit ripple carry adder for addition process.

The performance of 16-bit Vedic multiplier is increased in terms of area, delay and power when compared to performance of 16-bit array multiplier. We can easily design 32-bit and 64-bit Vedic multiplier with help of 16X16 and 32X32 bit Vedic multiplier. Hence from above consecution, it is clear that for large bit multiplication, Vedic multiplier gives more advantage than array multiplier.

However, there are some limitations in Vedic multiplier. In each NXN Vedic multiplier, three sets of ripple carry adders are essential for perform addition process of partial products. This multiplier's performance is better than array and modified booth multiplier's performance. Further to improve the performances of Vedic multiplier, different types of adders like Carry Save Adder (CSA), Carry Look-ahead Adder (CLA), Square Root Cary Select Adder (SQRT CSLA) is incorporated into Vedic multiplier. In order to reduce the limitation of Vedic multiplier, Wallace tree multiplier has been developed in later.

An adder is the main component of an arithmetic unit. Adders are commonly found in many building blocks of microprocessors and digital signal processing chip.

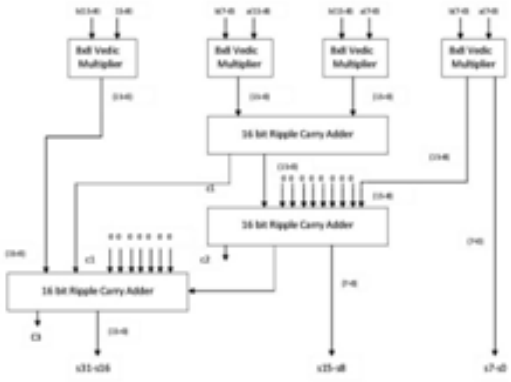

Fig. 2.9 8X8 Vedic Multiplier

Adders are essential not only for addition, but also for subtraction, multiplication and division. Addition is one of the fundamental arithmetic operations. A fast and accurate of digital system is greatly influenced by the performance of the resident adders.

The most important for measuring the quality of adder design were propagation delay and area. Application where these adders are used are multipliers, Digital signal processing to execute like fast Fourier transform (FFT), finite impulse response (FIR) and infinite impulse response (IIR).

2.3.6 Carry Select Adder

Design of area and power efficient high speed data logic systems are one of the most substantial areas of research in VLSI system design. Addition usually impacts widely the overall performance of digital systems and an arithmetic function. In electronics applications adders are most widely used. In multipliers, DSP to execute various algorithms like FFT, FIR and IIR. In digital adders, the speed of addition is limited by the time required to propagate a carry through the adder. The sum for each bit position in an elementary adder is generated sequentially only after the previous bit position has been summed and a carry propagated into the next position.

The CSLA is used in many computational systems to alleviate the problem of carry propagation delay by independent generation multiple carries and then select a carry to generate the sum. However, the CSLA is not an area efficient because it uses multiple pairs of Ripple Carry Adders(RCA) to generate partial sum and carry by considering carry input as Cin=0 and Cin=1, then the final summation and carry are selected by the multiplexers.

The Carry select adders are classified as Linear Carry select adder and Square-root Carry select adder.

III. SYSTEM DESIGN

PPROCESSING UNIT (MAC)

Multiplication and Accumulation (MAC) is one of the processing units in the neural network. Based on the performance of the MAC only, the accuracy of the network is obtained. MAC operation was performed, using the Vedic multiplier with SQRT-CSLA.

3.1.1 Vedic multiplier

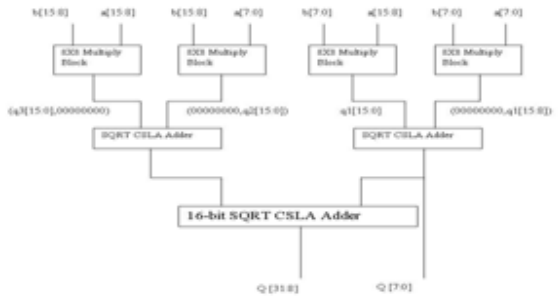

Fig. 3.1 Block Diagram of 16 X 16 Vedic Multiplier

Multiplication is one of the important arithmetic operations in signal processing applications. Signal processing involves multiplication, speed and accuracy is the main constraint in the multiplication process. Speed can be achieved by reducing the computation process in the multiplication

technique. Vedic multiplier is efficient multiplication technique.

Fig. 3.1 shows the architecture of 16-bit Vedic multiplier. It was designed by four 8 X 8 Vedic multiplier, each multiplier perform the operation separately. Partial products are added by 16-bit SQRT-CSLA; finally get a 32-bit multiplication output.

The efficient Vedic multiplication technique is used. The 16-bit Vedic multiplier is designed by using four 8X8 Vedic multiplier and square root carry select adder (SQRT-CSLA). The 16-bit input sequence is divided into two 4-bit numbers. Input to the 8-bit multiplier are a[7:0] & b[7:0], a[15:8] & b[7:0], a[7:0] & b[15:8], a[15:8] & b[15:8]. Intermediate partial products output are added using the three modified adder, named as SQRT-CSLA.

3.1.2 SQRT-CSLA Adder

Carry propagation delay and low complexity are recognized as high potential in every addition circuit. To achieve an efficient output, the proposed SQRT-CSLA structure has designed. SQRT-CSLA adder circuit is classified into two types based on selecting the carry inputs. a) Dual RCA based SQRT CSLA; b) BEC based SQRT CSLA.

In the dual RCA (Ripple Carry Adder) based SQRT CSLA circuit, each group has dual RCA pair for providing carry select signals. RCA circuit would be more disadvantageous due to the increasing propagation delay. To overcome the problem, Binary to Excess 1 converter circuit has been suggested in the SQRT-CSLA adder.

Fig. 3.2 shows the Architecture of BEC Based SQRT CSLA, it contain BEC, RCA and mux. Half adders, full adders and multiplexers are used for providing partial product addition results. BEC circuits are used to provide same RCA functions, but have different architectures with less gate count.

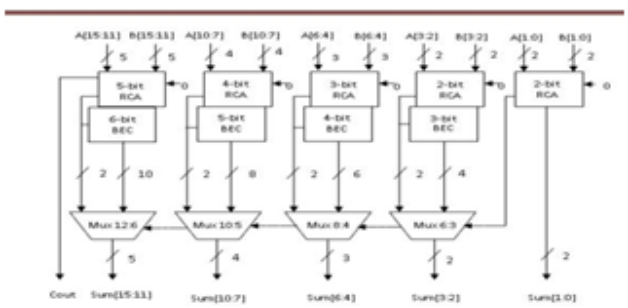

Fig. 3.2 Architecture of BEC based SQRT CSLA

3.1.3 Activation function unit

The Activation function of a single neuron in the artificial neural network is determined as function of the output in that neuron. Binary threshold neuron, Sigmoid neuron shown in Fig 3.3, and Rectified linear neuron are the different activation function used in the neural network. Output functions are varied by the activation function.

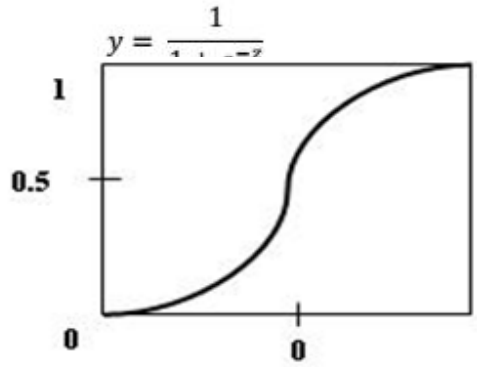

Fig. 3.3 Sigmoid Neuron

y is denotes the output function,
z is denotes as impulse response function.

The above equation describes the function of the sigmoid neuron. Sigmoid neuron is used as the activation function of the neural network. The above function is efficient compared to other activation functions.

3.2 SYSTEM ARCHITECTURE

The overall system architecture of the proposed work is presented in Fig 3.4. It consists of 4 different modules namely half sum generation, carry generation, carry selection and full sum generation.

All the redundant logic operation present in the conventional CSLA is eliminated and a new logic formulation is used in the proposed CSLA design.

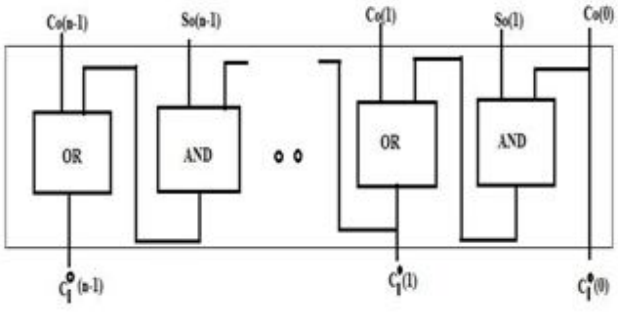

Fig. 3.4 System Architecture of Reduced Area-Delay-Power (ADP) - SQRT-CSLA

3.2.1 Half Sum Generation Unit

The Half Sum Generation units (HSG) receives the two n-bit operands (A and B) and generates the half sum word (S0) and half carry word (C0) of width n-bit each. The logic diagram of the HSG unit is shown in Fig 3.5. The Carry Generation unit receives both the half sum and half carry words from the HSG unit.

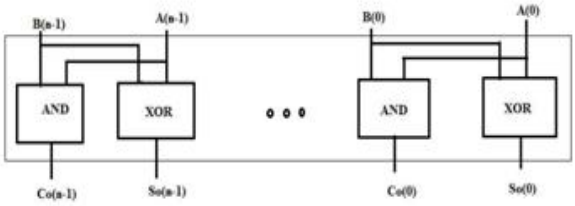

Fig. 3.5 Half Sum generation unit

3.2.2 Carry Generation Unit

The Carry Generation (CG) unit composed of two CGs (CG0 and CG1) corresponding to input carries 0 and 1. Both CG0 and CG1 receives half sum word (S0) and half carry word (C0) from the HSG unit and generate two n-bit full carry word C_1^0 and C_1^1 corresponding to the input carry 0 and 1 respectively. The logic circuits of CG0 and CG1 are optimized to take advantage of the fixed input carry bit. The optimized design of CG0 and CG1 are shown in Fig 3.6 and Fig 3.7 respectively.

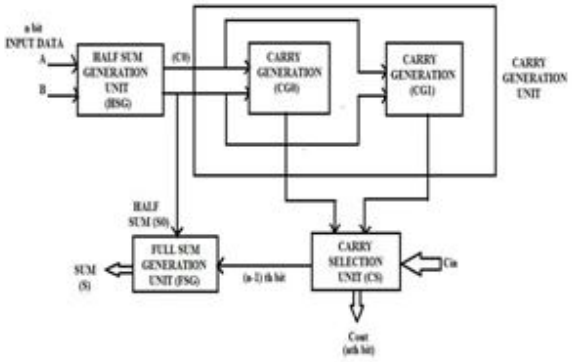

Fig. 3.7 Carry Generation Unit for Input Carry = 1

3.2.3 Carry Selection Unit

The Carry Selection (CS) unit selects one final carry word from the two carry words available at the input line using the control signal $C_{in.}$ It selects C10 when Cin = 0; otherwise it selects C_1^1. The CS unit can be implemented using an n-bit 2-to-1 MUX. However, the carry word C_1^0 and C_1^1 follow a specific bit pattern. If C_1^0 = 1 then C_1^1(i) = 1 irrespective of S0(i) and C0(i), for $0 \le i \le n-1$. This feature is used for logic optimization of Carry Selection unit. The optimized design of the CS unit is shown in Fig 3.8 which composed of n AND- OR gates.

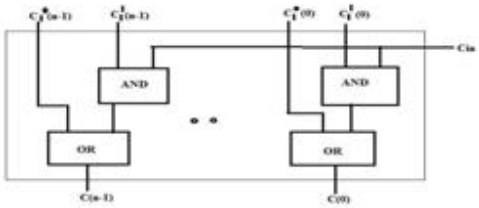

Fig. 3.8 Carry Selection Unit

3.2.4 Full Sum Generation Unit

The final carry word C is obtained from the carry Selection unit. The most significant bit (MSB) is sent to output as Cout, and (n-1) least significant bits are XORed with (n-1) MSBs of half sum (S0) in the full sum generation unit to obtain (n-1) MSBs of final sum (S). The LSB of S0 is XORed with Cin to obtain the LSB of S. The logic diagram of Full Sum Generation (FSG) unit is shown in Fig 3.9.

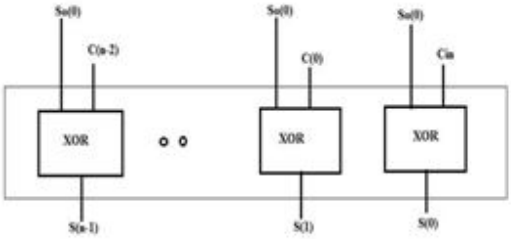

Fig. 3.9 Full Sum Generation Unit

IV. FLOW DIAGRAM

The flow chart shown in Fig 3.10 describes the generation of sum and carries word.

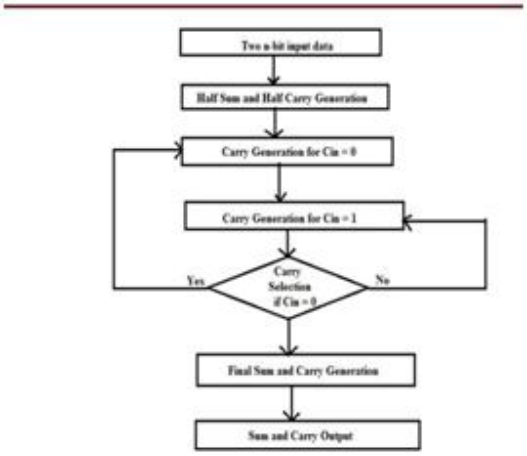

Fig. 3.10 Flow Chart of Reduced ADP- SQRT – CSLA

The HSG receives two n-bit input data and generate the half-sum word and half-carry word of width of n-bit each. Both CG0 and CG1 receives half sum and half carry word from the HSG unit and generate the two n-bit full-carry word corresponding to input carry 0 and 1 respectively. The CS unit selects one final carry word from the two carry word available at its input line using the control signal Cin.

The final carry word is obtained from the CS unit. The final sum is generated by the FSG unit.

V. RESULT OF VEDIC MULTIPLIER USING MODELSIM

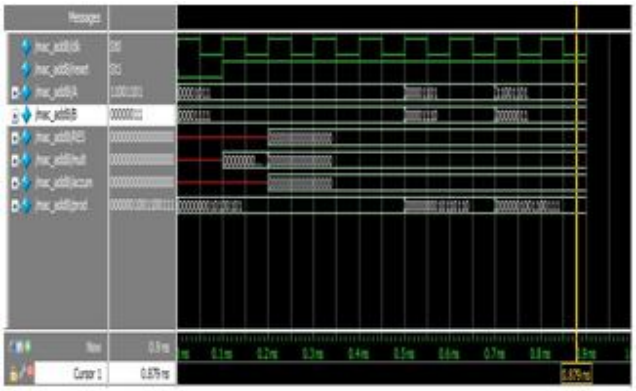

Fig. 6.2 MAC Unit Output

Fig 6.2 shows the output form of the MAC unit. Follow the below procedure to generate output wave form.

- Click on simulate button on top of ModelSim
- Work window will be popped up, select the Mac_add8 from work window and click OK button.
- Right click on Mac_add8 and select Add all signals to wave.
- Wave form window is generated which is shown in fig 6.2.
- Right click on clk and set it to clock.
- Right click on Reset and force reset value to 0.
- Right click on mac_add8/A and force value to 11001101 which is 205 in unsigned value.
- Right click on mac_add8/B and force value to 00000011 which is 3 in unsigned value.
- Click on Run button which is on top of the wave window.
- Output will be generated by multiplying 205 and 3 and comes output as 615.
- Right click on Reset and force it to 1.
- Click on Run button continuously Output will be generated by adding 205 to each cumulative output and will display in Accumulator line.

Table 6.1 Results of various multipliers

Parameters	Existing Booth Multiplier	Proposed Vedic Multiplier
LUT	764	717
Slices	402	397
Delay (ns)	19.114	19.1

From the above table it is clear that the Area, Delay and Power is reduced respectively for each Multipliers. It is clearly showing that numbers of LUTs, slices, Delay are reduced from existing booth multiplier and proposed vedic multiplier.

6.2 USING GRAPH

Fig 6.3 shows the graphical representation of MAC output using Booth multiplier and Vedic multiplier where it is clearly showing the difference in LUTs, number of slices and delay between Booth and Vedic multipliers.

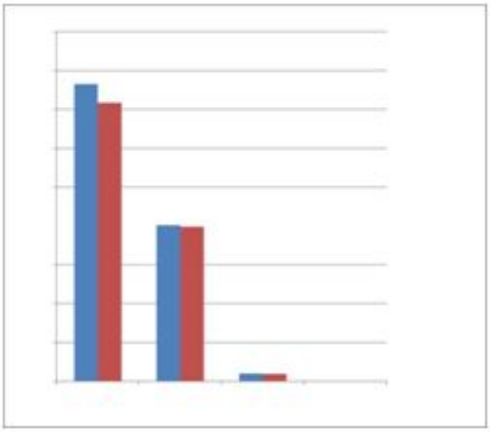

Fig. 6.3 Graphical Representation of MAC Unit Output

VI. CONCLUSION

Artificial Neural Networks are used in many applications, to analyze the methodology. MAC unit is one of the processing units in the artificial neural network. MAC decides the output function is efficient or not. So, a new MAC unit with the help of Vedic multiplier with SQRT-CSLA is designed. It produced the accurate and efficient output, compared to the existing booth multiplier with carry look ahead adder. Our proposed MAC increases the speed of the neural network. The MAC operation is performed well, entire network performance also increased.

In future the multiplier circuit is designed by using Reversible logic gates. It consumes less power compared to our ordinary logic gates. So this technique is applied to the neural network, get a better results.

REFERENCES

[1] R.NareshNaik , P.Siva Nagendra Reddy and K. Madan Mohan "Design of Vedic Multiplier for Digital Signal Processing Applications" in International Journal of Engineering Trends and Technology, volume 4 issue 7-2013 ISSN: 2231-5381(IJETT).

[2] P.Siva Nagendra Reddy et all.."Design and Implementation of FPGA based 64-bit MAC Unit using VEDIC Multiplier and Reversible Logic Gates" Indian Journal of Science and Technology, Vol 10(3),DOI:10.17485/ijst/2017/v10i3/109413, January 2017

[3] L. Ranganath, D. J. Kumar and P. S. N. Reddy, "Design of MAC unit in artificial neural network architecture using Verilog HDL," 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), Paralakhemundi, 2016, pp. 607-612. doi: 10.1109/SCOPES.2016.7955511

[4] Saman Razavi and Bryan A. Tolson, "A New formulation for feedforward neural networks" IEEE Transactions on neural networks, vol.22, October 2011.

[5] Richard L. et al. "Comparison of feedforward and feedback neural network architectures for short term wind speed prediction", International joint conference on neural networks, June 2009.

[6] Premananda B.S. et al. "Design and Implementation of 8-bit Vedic multiplier", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 2, Issue 12, December 2013.

[7] Damarla paradhasaradhi and K. Anusudha,"An area efficient SQRT carry select adder", International Journal of Engineering Research and Applications, vol. 3, Issue 6, Dec 2013.

[8] Hariprasath S and T.N. Prabakar," FPGA Implementation of multilayer feed forward neural network architecture using VHDL".

[9] S. Coric, I.Latinovic and A.Pavasovic," A Neural Network FPGA Implementation" ,IEEE, Neural-2000.

[10] G.Ganesh Kumar, V. Charishma,"Design of High speed Vedic multiplier using Vedic Research publication, volume 2, Issue 3, march 2012 mathematics Techniques", International Journal of scientific and

[11] K.Saranya,"Low power and area efficient carry select adder", International Journal of soft computing and Engineering, volume-2, Issue-6, January 2013.

[12] B. Ramkumar and Harish M kittur,"Low power and area efficient carry select adder", IEEE Transaction on Very large scale Integration (VLSI) systems, vol. 20.