MODELING OF PULSE-COUPLED NEUROHARDWARE USING SIMULINK©

M. NIRMALADEVI
Department of Electronics & Communication, Amrita School of Engineering,
Coimbatore, TamilNadu, India. 91- 0422 - 2656 422
E-mail Id: nirmala_amrita@yahoo.co.in, m_nirmala@ettimadai.amrita.edu

S. ARUMUGAM
Directorate of Technical Education, Chennai

Abstract: *Using Simulink© tool, the Complementary MOS (CMOS) modeling of Pulse-Coupled Neural Network is demonstrated. Compact architectures for computing XOR and Parity functions without the need for weight learning are explained. Any N-bit parity function can be simulated with N hidden layer neuron, by the conventional method. As N increases, the network size grows and becomes complex for Very Large Scale of Integration realization. So, the fully connected networks are tried, which needs only N/2 neurons in the hidden layer. The architectures are claimed to be suitable for hardware implementation, since the majority of weights are equal to +1 that obviates the need for multiplier. Finally the network is configured to solve Character Recognition application. It also exhibited a satisfactory performance accepting 10% noisy patterns.*

Keywords: *Pulse Coupled Neural Network, Neurohardware, Simulink, VLSI*

1. Introduction

Artificial Neural Networks (ANN) are built up from interconnected objects, which are simplified representations of biological neurons. They provide a good solution for the computation intensive problem like pattern classification. Data from neurobiological experiments have made it increasingly clear that biological neural networks, which communicate through pulses, use the timing of the pulses to transmit information and perform computation. Here, communication is done using frequency modulated pulse streams [1]. The firing rate of action potentials is roughly proportional to change in the original graded potential, which is categorized as frequency modulation (FM). Neurobiological experiments have stimulated significant research on Pulsed Neural Networks varying from theoretical analysis to hardware implementation [2].

Hardware implementation of ANN shows inherent fault tolerance specialties and high speed, which is usually more than an order of magnitude over the software counterpart. More and more ANN designs are carried out in analog, digital, mixed and optical circuits in Very Large Scale of Integration (VLSI) hardware because their regular computation and communication structures makes them a good match for custom VLSI implementations [3]. Pulse Stream (PS) techniques for VLSI neural networks have been developed, to incorporate the merits of both digital and analog technology. PS implementations have advantages such as, higher noise immunity, smaller size, easier to multiplex, interface and reconfigure.

Many researchers have tried different pulse modulation schemes. G. Moon et al. [4] designed neuron cell that encodes the information into the form of pulse duty cycles. J. Meador et al. [5] designed FM pulse firing circuits. Review of Pulse Coupled Neural Network (PCNN) models is presented in [6, 7] and Padgett et al. [8] presented the review of PCNN for Pattern recognition applications.

The Pulse Coupled Neuron (PCN) design adapted in this paper follows natural biological process that utilizes FM [9]. It is a simple model without the need for multiplier and can be implemented in low voltage mode operation [10]. The PCNN is not extremely complicated so it does not require extensive mathematical skills. Using PCNN, two-bit XOR, N-bit parity and Character recognition applications are realized. The hardware design uses a leaky integrate-and-fire pulse generator with adaptive synaptic coupling. Neuron's coupling strength determines the synchronization effect, which is useful in pattern and character recognition. Architectural and functional descriptions, followed by the analysis of simulated results are presented.

2. CMOS Architecture of Pulse-Coupled Neuron

Design of a voltage mode pulse-coupled neuron using MOSFET with synaptic multiplication and summation is described in this section. Metal Oxide Semiconductor FETs (MOSFETs) are the active elements preferred in VLSI design of circuits.

Instead of using any conventional tool, the Simulink tool is used to model the MOSFETs and hence the neuron. The work presented here, shows a significant improvement in research, when compared to a few work reported [11, 12]. A versatile MOSFET is modeled which could be utilized to realize different architectures.

2.1 MOSFET Modeling

Neuron model uses n-channel (nMOS) and p-channel (pMOS) FETs. Both function in linear and saturated mode of operation depending upon the applied Gate-Source Voltage V_{GS} and conduct a Drain Current I_D represented by the equations (1), (2) and (3) given below. The small-signal equivalent circuit with its Simulink model and the characteristic curve are shown in Fig. 1 and 2 respectively. The PCN is modeled with MOSFETs after verifying the performance.

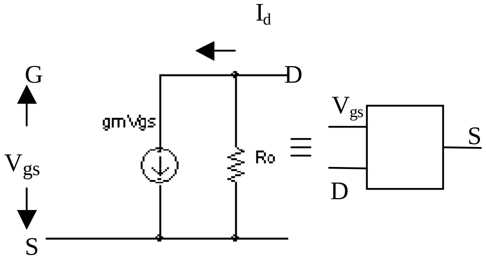

Fig. 1. Small Signal Model of nMOS

In Cut-off region, $V_{GS} < V_T$; $I_D = 0$ (1)

In Linear region, $V_{GS} > V_T$ & $V_{DS} < V_{GS} - V_T$;
$I_D =(W/L)\ \mu_n C_{ox}[(V_{GS} - V_T)V_{DS} - (V_{DS}^2/2)]$ (2)

In Saturation region, $V_{GS} > V_T$ & $V_{DS} >= V_{GS} - V_T$;
$I_D = I_{Dsat} =(W/ 2L)\mu_n C_{ox}(V_{GS}-V_T)^2[1+\lambda(V_{DS})]$ (3)
Where,

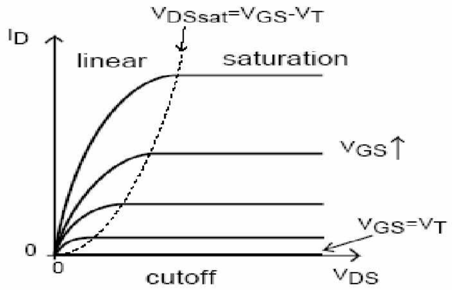

Fig. 2. Characteristic Curve of nMOS

V_{GS} – Gate-Source Voltage; V_T – Threshold Voltage; I_D – Drain Current; V_{DS} – Drain-Source Voltage;
W, L – Width and Length of MOS; μ_n – Mobility of electrons; C_{ox} – Oxide Capacitance;
λ - *Channel length modulation co-efficient.*

2. 2 Structure and Function of Pulse-Coupled neuron

The Neuron shown in Fig.3 is an electronic analogy of a biological neuron [9].

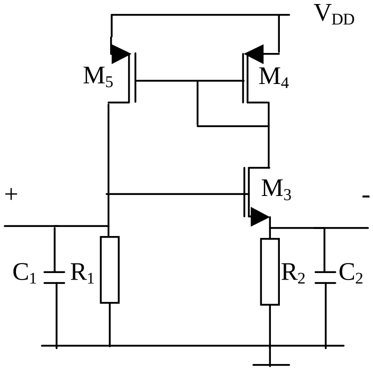

Fig. 3. Model of a Pulse-Coupled neuron

When an external stimulus is applied, it initiates reactions by generating a pulse stream. The Neuron uses two capacitors with different time constants. Capacitor C_1 is used to integrate the incoming signals and C_2 is to control the threshold voltage of M_2 and refractory period after the pulse stream is generated. R_1 and R_2 together with C_1 and C_2 determine the time constants. The charge across C_1 represents the charge of Sodium ions (Na+), and its potential should change at a faster rate than the potential due to Potassium ions stored across C_2. Hence the following equation (4) is to be satisfied.

$$R_1C_1 < R_2C_2 \quad (4)$$

Passive resistors are not preferred in VLSI fabrication. Hence diode-connected transistors that realize active resistors are used. Modified circuit is shown in Fig. 4.

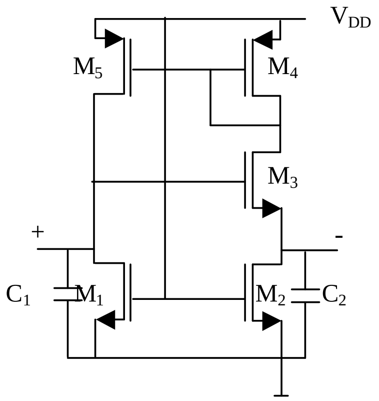

Fig. 4. PCN with active resistors

Designed neuron has two input nodes, one at each capacitor, to control both excitatory and inhibitory signals. Initially the MOS transistors M_3, M_4 and M_5 are cut-off, when not triggered. Assume both positive and negative inputs are applied. Now C_1 charges faster than C_2 and at a point of time, gate voltage of M_3 exceeds its source voltage by V_t, which turns M_3 on. It triggers M_4 and M_5 on which is a current mirror. The neuron remains in this state, until the charge across C_2 is insufficient. When charge of C_2 exceeds threshold voltage, it turns M_3 off and hence M_4 and M_5 are turned off. Now refractory period is setup and the neuron does not respond to any excitations until the potential on C_1 exceeds the potential on C_2 by the threshold value of M_3.

Positive inputs applied to C_1, helps to trigger the transistor M_3 fast. At the same time, the negative input applied to C_2, increases the threshold of M_3 and hence slows down the triggering. The neuron excites, when the potential due to positive input exceeds the potential due to negative input. Hence the natural phenomena- Integrate- and- Fire of biological neuron is exhibited. Simulink Model of a PCN and its symbol is shown in Fig. 5. To realize a complete network, the neurons are to be connected together, with differing synaptic weights.

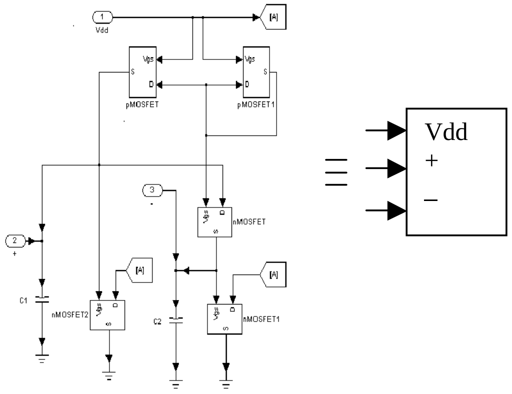

Fig. 5. Simulink Model and Symbol of a PCN

3. Design Implementation

Networks to realize the functions of two-bit XOR, N-bit Parity and Character recognition are implemented using the PCN explained in Section2. Details of applications are discussed in the following subsections.

3.1 Two-bit XOR Function

In the elementary (single layer) perceptron, there are no hidden neurons. Consequently, it cannot classify input patterns that are not linearly separable. However, nonlinearly separable patterns are of common occurrence. Two-input XOR is one of the simplest non-trivial and linearly inseparable problems. It needs a three-layered network with one hidden layer. Digital phase detectors and Code converters are realized using XOR modules. N-bit parity generator and checker is also designed using XOR blocks. Fig. 6 shows XOR function with PCN [9].

3.1.a XOR Function using PCN

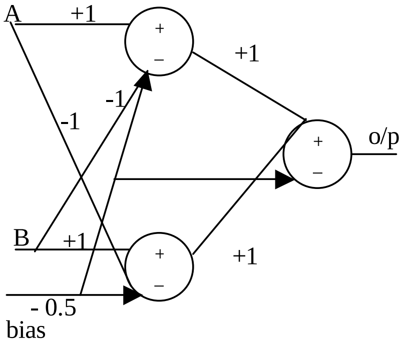

Fig. 6. XOR function using PCN

3.1.b XOR Function with unipolar neurons

Design of two-bit (*a*, *b*) XOR using unipolar neuron [13] is done without learning the weight values. Instead, the following expressions are used to realize the function.

$$a + b - 0.5 > 0 \quad (5)$$
$$a + b - 1.5 > 0 \quad (6)$$

Equation (5) is satisfied if any one input is high (i.e) +1. Equation (6) is met when both inputs are high. The following observations are made from the above equations.

i. All weights from input to hidden layer is +1
ii. Biasing weights are (-0.5, - 1.5) respectively
iii. Weights for the output neurons are +1, -1 and the bias is -0.5.

Fig. 7 shows XOR using unipolar neurons, where N_H and N_O indicate hidden and output layer neurons respectively.

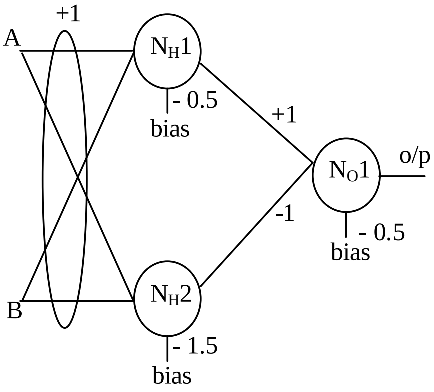

Fig. 7. XOR using unipolar neurons

3.2. N-bit Parity Function

Parity circuits play a vital role in communication for error detection and correction. Error detection in digtal memory is done using parity checks. Realization of parity-N generator using XOR needs (N-1) XOR gates. It is implemented in several layers and hence introduces significant delay. Instead a simple and straightforward technique is followed in this paper, to realize N-bit parity functions.

Many researchers have solved N-bit Parity Functions using neurons with specific activation functions [14]. To solve N-bit Parity Functions, the minimum number of hidden layer neurons is N [15]. In a fully connected network, the number of neurons in hidden layer reduces to N/2. [16]. The total number of neurons to solve N-bit Parity Function problem is further reduced to $\log_2$N. [17] MATLAB results of different architectures were presented in [13]. As a verification-of-concept, the CMOS realization of N-bit Parity Function is demonstrated using specially modeled pulse-coupled neurons. Hardware realization is further simplified, since the majority of weights equal +1, and hence weight multiplication is not needed.

3.2.a Three-bit Parity Function

To solve three *(a, b, c)*-bit Parity Function using unipolar neurons, the equations are given below

$$a + b + c - 0.5 > 0 \quad (7)$$
$$a + b + c - 1.5 > 0 \quad (8)$$
$$a + b + c - 2.5 > 0 \quad (9)$$

Equation (7) is satisfied if at-least one input is high. Equation (8) is met when any two inputs are high. Equation (9) is satisfied if all inputs are high. The following specifications are derived using the above equations.

iv. All weights from input to hidden layer is +1
v. Biasing weights are (-0.5, -1.5, - 2.5) respectively
vi. Weights for the output neurons are +1, -1, +1 and the bias is -0.5

Direct realization of network is done without the need for weight learning. Architecture of three-bit parity function is shown in Fig. 8.

Any N-bit parity function with N hidden layer neuron can be simulated. As N increases, the network size grows and becomes complex for hardware realization. So, the fully connected networks are tried, which needs only N/2 neurons in the hidden layer.

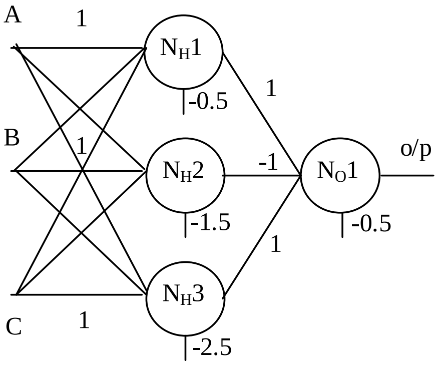

Fig. 8. Three-bit Parity Function

3.2.b Four-bit Parity Function

For four-bit parity functions, MATLAB simulated architecture was 4:7:1. It is reduced to 4:2:1 when fully connected architecture is simulated. Parity N functions are symmetrical. i.e. the output is generated based on the number of excited inputs, and the position of input does not influence it. Model of Fully connected network is shown in Fig. 9.

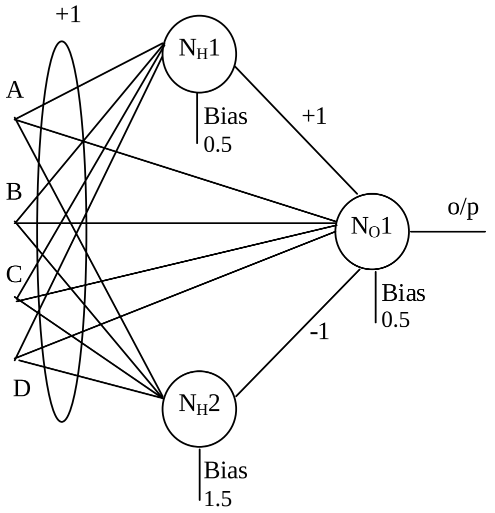

Fig. 9. Four-bit Parity Function

3.3 Character Recognition

Character recognition is a trivial task for humans, but for computers it is extremely difficult. The main reason for this is, the many sources of variability. There exist several different techniques for recognizing characters. One distinguishes characters by the number of loops in a character and the direction of their concavities. In terms of recognition and feature extraction, PCNN can be very effective [18]. Simulink model and results of all the applications are discussed in Section 4.

4. Experimental Results

4.1 Two-bit XOR

4.1.a Using PCN

By mapping the inputs and weights, the functionality of PCN is tested. Simulink model is shown in Fig. 10.

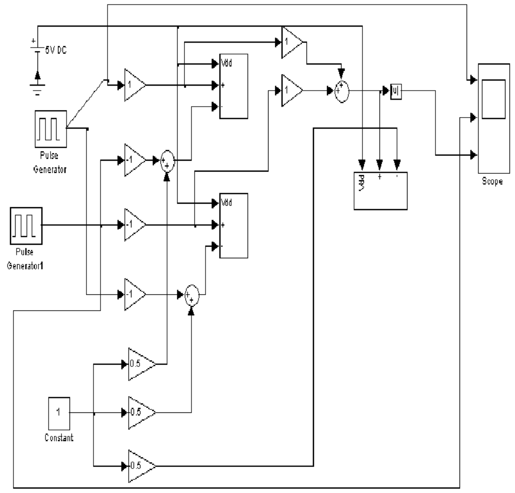

Fig. 10. Model of XOR Using PCN

4.1.b. XOR using unipolar neuron

Fig. 11 shows the model of XOR Using Unipolar Neurons.

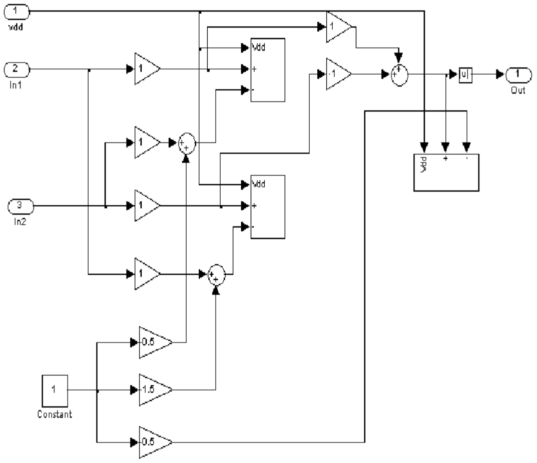

Fig. 11. Model of XOR Using Unipolar Neurons

XOR results are presented with two input waveforms and one output waveform in Fig. 12.

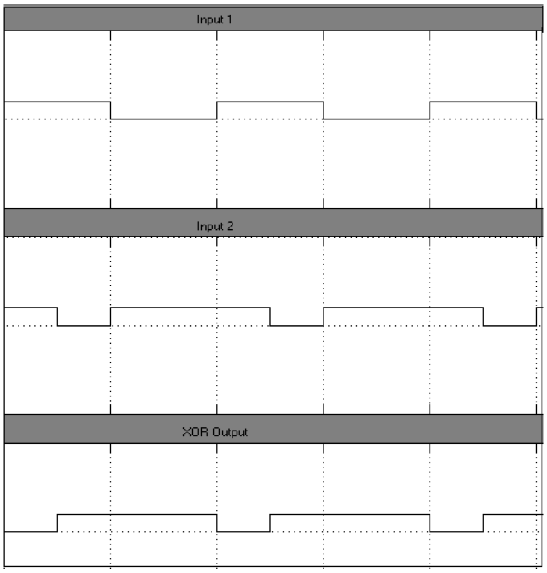

Fig. 12. Simulated Results of XOR

4.2 N-bit Parity function

4.2. a. Three-bit parity

Model and results of three-bit Parity function are presented in Fig. 13 and 14 respectively.

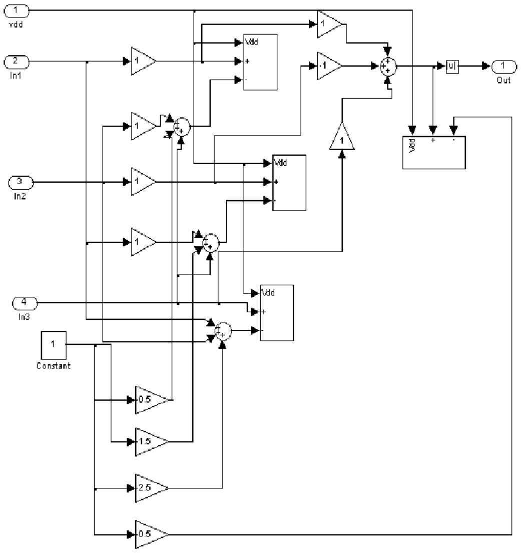

Fig. 13. Model of three-bit parity

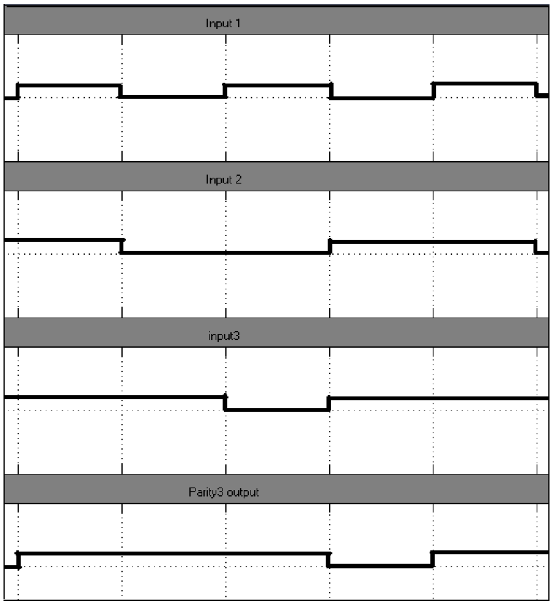

Fig. 14. Results of three-bit parity

4.2. b. Four-bit parity

Fully connected network to solve four-bit parity function with only two hidden layer neurons and its simulated results are shown in Fig. 15. and Fig. 16 respectively.

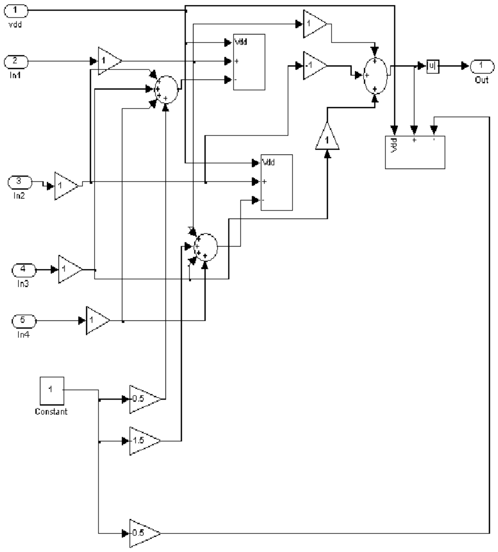

Fig. 15. Model of four-bit parity

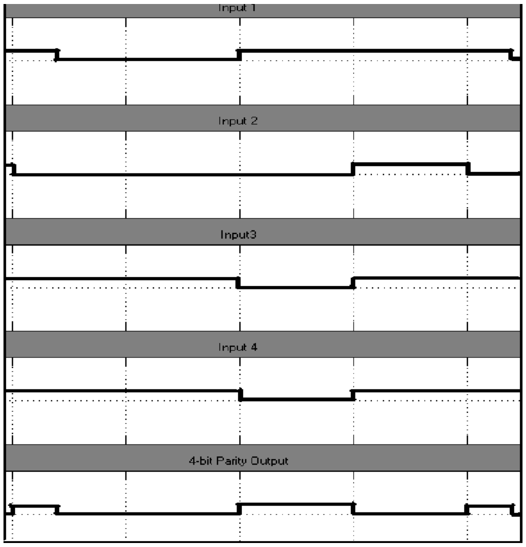

Fig. 16. Results of Four-bit Parity

4.3 Character Recognition

CMOS model of PCN is used to configure the character recognition application. The alphabets A to Z in 7 × 5 matrix are encoded as 32-bit data stream. The output stage is modeled with 26 display devices in Simulink. Applied input stream triggers the corresponding display to indicate '1'. For e.g. if the character 'S' is applied, nineteenth display is excited. Block diagram of the arrangement is shown in Fig. 17.

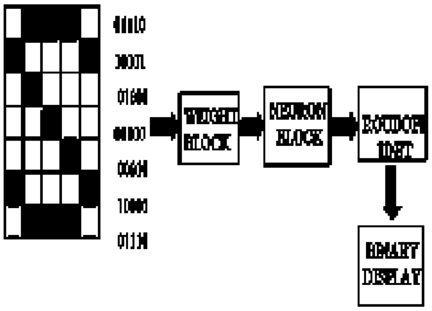

Fig. 17. Block diagram of Character Recognition

Here, each character is represented by 7 X 5 pixels, forming an array of 35 input stimuli. The input data is multiplied with the appropriate weight in this block.

Totally 35x26 weights are added in this block. The weights are obtained by using the back propagation algorithm. As an example command for Character 'J' is shown.

```
s=[0;alphabet(:,10)]';
s = Columns 1 through 18
0 1 1 1 1 1 0 0 1 0 0 0 0 1 0 0 0 0
Columns 19 through 36
1 0 0 0 0 1 0 0 1 0 1 0 0 0 1 0 0 0
```

Display '10' is excited to display '1' as shown in Fig. 18.

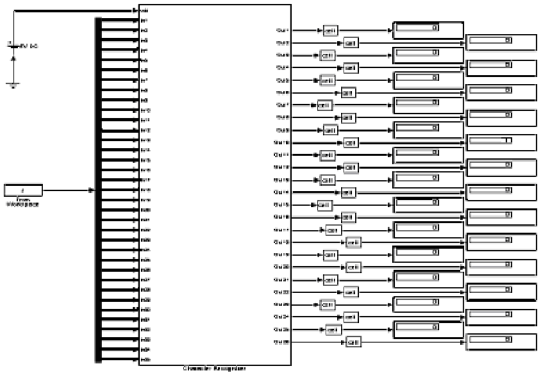

Fig. 18. Architecture of Character Recognition

5. Conclusion

The PCNN discussed is modeled in CMOS technology using Simulink tool. The simple and compact design reduces the circuit complexity. Different neural network architectures for Two-bit XOR and N-bit Parity applications are implemented with PCNN and unipolar neurons. Two-bit XOR is realized in two different methods. The first design uses PCN with trained weight and threshold. Second design uses uniploar neuron with biasing weights found from equations and learning was not needed. For Parity function, the feedforward network with one hidden layer requires N neuron in the hidden layer. It is reduced to N/2 in a fully connected network. As a verification-of-concept, a three-bit parity using unipolar neuron and four-bit parity using fully connected network is tested. Performance of PCNN in different applications is found to be satisfactory. The architectures are claimed to be suitable for hardware implementation, since the majority of weights are equal to +1 that obviates the need for multiplier. Character Recognition application also exhibited a satisfactory performance accepting 10% noisy patterns. Research on higher bit parity function and cascaded networks with unipolar neurons are being carried out.

References

1. Ota Y. and Wilamowski B. M. : *CMOS Implementattion of a Pulse-Coded Neural Network with a Current Controlled oscillator*, In: Proceedings of IEEE Int. Symp. Circuits and Systems, No. 3, 1996.
2. Wolfgang M, Christopher M. : Bishop, *Pulsed Neural Networks*, Cambridge, MA: MIT Press, 1999.
3. Reyneri L. M. et. al : *A Comparison between Analog and Pulse Stream VLSI hardware for Neural Networks and Fuzzy Systems,* In: Proceedings of IEEE Conf. 1994, pp. 77-86.
4. Moon et. al, *VLSI Implementation of Synaptic Weighting and Summing in Pulse Coded neural –type cells,* In: IEEE Trans. Neural Networks, (1992), Vol. 3, No. 3, May 1992. pp. 394-403.
5. Meador J. L. et.al : *Programmable Impulse Neural Circuits*, In: IEEE Trans. Neural Networks, (1991),Vol. 2, No. 1, Jan 1991. pp. 101-109.
6. Zaghloul M. E., Meador J. L., Newcomb R. W.: *Silicon Implementation of Pulse coded Neural Networks*, Kluwer, Boston, 1994.
7. Murray A. F. and Tarassenko L.: *Analog Neural VLSI: Pulse stream Approach*, Chapman &Hall, London, UK, 1994.
8. Padgett M. L., Webros P. J. and Kohonen T.: *Strategies & Tactics for the Application of Neural Network in Industrial. Electronics*, CRC Handbook for Indl. Electronics, CRC press and IEEE press, 1997, pp. 835-857.
9. Ota Y. and Wilamowski B. M. : *Analog Implementtaion of pulse-coupled Neural Networks,* In: IEEE Trans. Neural Networks, (1999),Vol. 10, No. 3, May 1999. pp. 539-544.
10. Pan H., Manic M., Li X., Wilamowski B. M.: *Multilevel Logic Multiplier Using VLSI Neural Network,* In: Proceedings IEEE- ICIT – 2003, Maribor, Slovenia, pp. 327-332.
11. Nassar K., Voorhees K. and Blaise J. H., *Mathematical Modeling of an Active Neural Network Using MATLAB and Simulink,* In: Proc IEEE Conference, 2006, pp. 115-116.
12. Yu Y. and Murakami K., *Reconfigurable Neural Network Using DAP/DNA*, In: Proceedings of the Seventh International Conference, HPCAsia'04, 2004
13. Wilamowski B. M., Hunter D. and Malinowski A.: *Solving Parity-N Problems with Feedforward Neural Networks*, In: Proceedings of Int. Joint Conf. on Neural Networks, Vol. IV, 2003, pp. 2546-2551.
14. Donald J. and Akers L. A. : *A Neural Processing node with on-chip Learning*, In: Proceedings of IEEE, Int. Conf Circuits and Systems, Chicago, 1993, pp. 2748.
15. Kim Y.-C.and Shanblatt M. A.: *Architecture and Statistical Model of a Pulse-mode Digital Multilayer Neural Network,* In: IEEE Trans. Neural Networks, (1995), Vol. 6, Sept. 1995. pp. 1109-1118.
16. Minnick R. C.: *Linear Input Logic*, In: IRE Trans. Electronics, Vol. EC-10, Mar. 1961, pp. 6-16.
17. Cotofana S. and Vassiliadis S.: *Periodic Symmetric Functions, Serial Additions and Multiplications with Neural Networks,* In: IEEE Trans. Neural Networks, (1998), Vol. 9, Nov. 1998, pp. 1118-1128.
18. Wilamowski B. M. et. al. : *Pulse-coupled neurons for image filtering,* In: Proceedings of World Congress of Neural Networks, SanDiego, CA, Sept. 15-20, 1996, pp. 851-854.

M. NirmalaDevi received her B. E. degree in Electronics and Communication Engineering and M. E. degree in Applied Electronics from the Bharathiar University, Coimbatore, India in 1990 and 1996 respectively. And she is currently doing Ph.D at Anna University, Chennai, India. Her area of research includes Artificial Neural Networks and VLSI Design. She has authored over 10 papers in National and International Conferences and Journals. She is a member of ISTE , IETE and VLSI society of India.

Dr. S. Arumugam, received the Ph.D degree in Computer Science and Engineering from Anna University, Chennai in 1990. He also obtained his B.E. (Electrical and Electronics Engineering) and M.Sc. (Engg.) (Applied Electronics) degrees from P.S.G College of Technology, Coimbatore, at University of Madras in 1971 and 1973 respectively. He has held various positions from associate lecturer, lecturer, assistant professor, professor and principal from 1974. Presently, he is working as Additional Director in the Directorate of Technical Education, Government of Tamil Nadu. He has guided four Ph.D scholars and guiding 10 Ph.D scholars. He has published 70 technical papers in international and national journals and conferences. His area of interest includes network security, biometrics and neural networks.