AMSE JOURNALS-2016-Series: Advances D; Vol. 21; N°1; pp 66-77
Submitted July 2016; Revised Oct. 11, 2016, Accepted Nov. 15, 2016

Design of a CMOS "OR Gate" using Artificial Neural Networks (ANNs)

R. K. Mandal

Department of Computer Science & Application, University of North Bengal
Raja Rammohunpur, PO: NBU, Distt: Darjeeling, West Bengal – 734013 India
(rakesh_it2002@yahoo.com; http://www.nbu.ac.in)

Abstract: **-** This paper is an approach to simplify the electronic circuits by using Complementary Metal Oxide Semiconductors (CMOS) transistors and map those to equivalent Artificial Neural Networks (ANNs). The implementation of those circuits in ANN leads to software implementation of complex circuits which makes them cheaper as compared to their hardware counterpart. In this paper a multiple layer ANN is developed for the OR gate using already designed CMOS OR gate. The transistors used in CMOS are replaced by simple artificial neurons. The weights used in this ANN are fixed and negative weights are considered for the inverters.

Key-words: - Complementary Metal Oxide Semiconductors (CMOS), Artificial Neural Networks (ANNs), OR Gate

I. Introduction

The modern computing is more biased towards intelligent computing. Intelligence can be best incorporated in computers using Artificial Neural Networks (ANNs) [1, 2]. Research work is going on to develop models which can be used in vast range of applications like medical informatics, handwriting recognition, speech recognition, and other applications of pattern recognition [3, 4, 5, 6]. The development of these models leads to implementation of more

complex circuits in software by keeping the hardware simple. These make these circuits cheaper. If ANNs are developed equivalent to the CMOS circuits, then it becomes simple to map ANNs to hardware circuits using CMOS [7, 8, 9, 10, 15].

Work has been done by Forssell M in the field of hardware implementation of Artificial Neural Networks [11]. Some work has already been done in this field where CMOS circuits were designed which accepts synaptic inputs and generates pulse width modulated output waveform of constant frequency on the basis of activation level [12]. Logic gates were implemented in single layer and two layers feed forward neural network based on supervised learning [13]. In another approach Artificial Neural Network (ANN) was used to demonstrate the way in which the biological system was processed in analog domain by using analog component as Gilbert cell multiplier, Adder, Neuron activation function for implementation [14]. Hui W et al worked on the use of artificial neural networks on segmented arc heather failure prediction [16].

This paper has been divided into three sections. Section-1 discusses the implementation of a simple ANN CMOS OR gate. Section-2 discusses the methodology. Section-3 discusses the result analysis.

II. Design of CMOS OR gate using Artificial Neural Networks (ANN) [15]

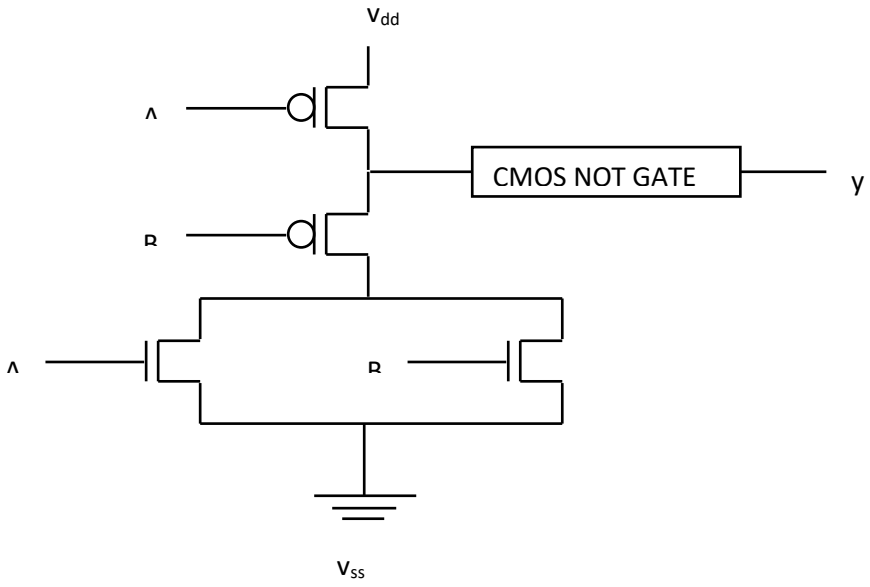

Figure 1: CMOS OR GATE

Figure 1 [15], displays a CMOS OR gate which is divided into two portions. The upper portion of the CMOS gate consists of two transistors connected in series. The upper portion is P-MOS. It is also called Pull up (PUN). The transistor inputs are A and B which are inverted before going to the transistor. The value of 'v_{dd}' is always 1. The lower portion of the CMOS gate consists of two transistors connected in parallel. The lower portion is N-MOS. It is also called Pull down (PDN). The transistor inputs are A and B. The value of 'v_{ss}' is always 0. When A=0 and B=0, the upper transistors connected in series gets inputs as 1. This clears the line for the 'v_{dd}'. The output obtained here will be 'v_{dd}', which is 1 and an inverted output is obtained because a NOT gate is connected further. In case of lower transistors the 0 input in both parallel transistor blocks the way for 'v_{ss}'. When A=0 and B=1, the first upper transistor gets input as 1 and the second upper transistor gets the input as 0 and blocks the way for 'v_{dd}'. The first lower transistor gets 0 and blocks the way for 'v_{ss}'. The second lower transistor gets output as 1 and clears way for 'v_{ss}'. The output obtained here will be 'v_{ss}'. Similarly, other outputs obtained are the outputs of an OR gate as shown in Table 1, [15].

Table 1: Truth Table of C-MOS OR GATE

A	B	v_{dd}	v_{ss}	y'	y
0	0	1	0	1	0
0	1	1	0	0	1
1	0	1	0	0	1
1	1	1	0	0	1

The OR gate already designed by using transistors and inverters are designed using a multiple layer Artificial Neural Network (ANN) as shown in Figure 2.

III. Methodology of designing CMOS OR gate using Artificial Neural Networks (ANN)

In case of an equivalent ANN circuit. The transistors are replaced by simple artificial neurons. The inverters are replaced by negative weights used on the links. Multiple layers of neurons are used to implement the ANN circuit.

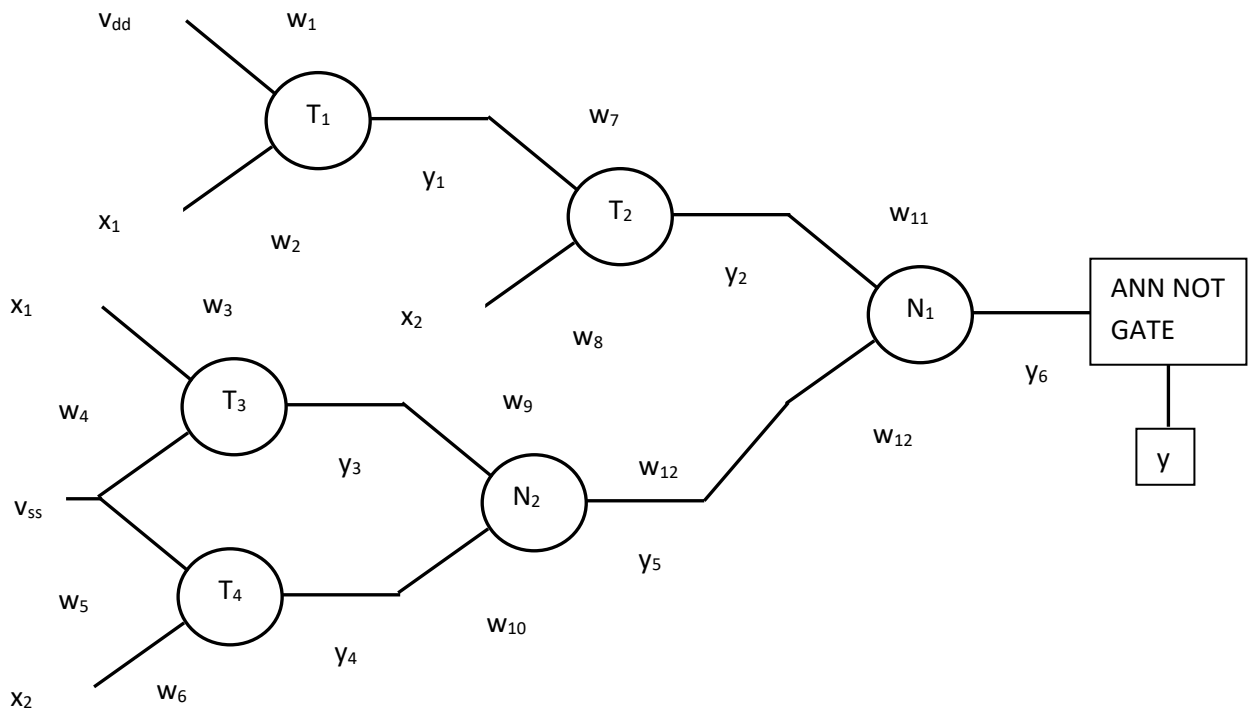

Figure 2: C-MOS OR Gate using multiple-layer ANN

Figure 2 shows a multiple layer ANN, where T_1 and T_2 are the artificial neurons representing the two transistors connected in series. And the neurons T_3 and T_4 represent the transistors connected in parallel. N_1 and N_2 are two simple neurons. One neuron is a simple perceptron and another neuron is a referee neuron. The weighted (w_1 and w_2) inputs of T_1 are v_{dd} and x_1 and output is y_1. The weighted (w_7 and w_8) inputs of T_2 are y_1 and x_2 and output is y_2. The weighted (w_3 and w_4) inputs of T_3 are x_1 and v_{ss} and output is y_3. The weighted (w_5 and w_6) inputs of T_4 are v_{ss} and x_2 and output is y_4. The weighted (w_{11} and w_{12}) inputs of N_1 are y_2 and y_5 and output is y_6. The weighted (w_9 and w_{10}) inputs of N_2 are y_3 and y_4 and output is y_5. Final output of the ANN is connected to a CMOS NOT gate.

The output of the neurons T_1, T_2, T_3, T_4 and N_2 are calculated by using the following equations:

$y_{out1} = v_{dd}*w_1 + x_1*w_2$ Equation 1

$y_{out2} = y_1*w_7 + x_2*w_8$ Equation 2

$y_{out3} = x_1*w_3 + v_{ss}*w_4$ Equation 3

$y_{out4} = v_{ss}*w_5 + x_2*w_6$ Equation 4

$y_{out5} = y_3*w_9 + y_4*w_{10}$ Equation 5

for (j=1 to 5; if $y_{outj} > 0$; then $y_j = 1$ else $y_j = -1$) Equation 6

The output of the neuron N_1 is calculated by using the following equation:

If ($y_2 == 1$ && $y_5 == -1$) then $y_6 == v_{dd}$ Equation 7

Example 1: The following example calculates the inputs and outputs of the CMOS OR gate using ANN.

$x_1 = -1$ and $x_2 = -1$

$y_{out1} = v_{dd}*w_1 + x_1*w_2$

$= 1*1 + (-1)*(-1)$

$= 2 > 0$

Therefore, $y_1 = 1$

$y_{out2} = y_1*w_7 + x_2*w_8$

$= 1*1 + (-1)*(-1)$

$= 2 > 0$

Therefore, $y_2 = 1$

$y_{out3} = x_1*w_3 + v_{ss}*w_4$

$= (-1)*1 + (-1)*1$

= -2 < 0

Therefore, $y_3 = -1$

$y_{out4} = v_{ss}*w_5 + x_2*w_6$

= (-1)*1 + (-1)*1

= -2 < 0

Therefore, $y_4 = -1$

$y_{out5} = y_3*w_9 + y_4*w_{10}$

= (-1)*1 + (-1)*1

= -2 < 0

Therefore, $y_5 = -1$

Applying Equation 7 we get $y_6 = 1$

Therefore, y = 0

x_1 = -1 and x_2 = 1

$y_{out1} = v_{dd}*w_1 + x_1*w_2$

= 1*1 + (-1)*(-1)

= 2 > 0

Therefore, $y_1 = 1$

$y_{out2} = y_1*w_7 + x_2*w_8$

= 1*1 + 1*(-1)

= 0

Therefore, $y_2 = -1$

$y_{out3} = x_1*w_3 + v_{ss}*w_4$

= (-1)*1 + (-1)*1

= -2 < 0

Therefore, $y_3 = -1$

$y_{out4} = v_{ss}*w_5 + x_2*w_6$

= (-1)*1 + 1*1

= 0

Therefore, $y_4 = -1$

$y_{out5} = y_3*w_9 + y_4*w_{10}$

= (-1)*1 + (-1)*1

= -2 < 0

Therefore, $y_5 = -1$

Applying Equation 7 we get $y_6 = -1$

Therefore, y = 1

$x_1 = 1$ and $x_2 = -1$

$y_{out1} = v_{dd}*w_1 + x_1*w_2$

= 1*1 + 1*(-1)

= 0

Therefore, $y_1 = -1$

$y_{out2} = y_1*w_7 + x_2*w_8$

= (-1)*1 + (-1)*(-1)

= 0

Therefore, $y_2 = -1$

$y_{out3} = x_1*w_3 + v_{ss}*w_4$

= 1*1 + (-1)*1

= 0

Therefore, $y_3 = -1$

$y_{out4} = v_{ss}*w_5 + x_2*w_6$

= (-1)*1 + (-1)*1

= -2 < 0

Therefore, $y_4 = -1$

$y_{out5} = y_3*w_9 + y_4*w_{10}$

$= (-1)*1 + (-1)*1$

$= -2 < 0$

Therefore, $y_5 = -1$

Applying Equation 7 we get $y_6 = -1$

Therefore, y = 1

$x_1 = 1$ and $x_2 = 1$

$y_{out1} = v_{dd}*w_1 + x_1*w_2$

$= 1*1 + 1*(-1)$

$= 0$

Therefore, $y_1 = -1$

$y_{out2} = y_1*w_7 + x_2*w_8$

$= (-1)*1 + 1*(-1)$

$= -2 < 0$

Therefore, $y_2 = -1$

$y_{out3} = x_1*w_3 + v_{ss}*w_4$

$= 1*1 + (-1)*1$

$= 0$

Therefore, $y_3 = -1$

$y_{out4} = v_{ss}*w_5 + x_2*w_6$

$= (-1)*1 + 1*1$

$= 0$

Therefore, $y_4 = -1$

$y_{out5} = y_3*w_9 + y_4*w_{10}$

$= (-1)*1 + (-1)*1$

$= -2 < 0$

Therefore, $y_5 = -1$

Applying Equation 7 we get $y_6 = -1$

Therefore, $y = 1$

IV. Result Analysis

Table 2 shows various inputs given to different artificial neurons used in the CMOS OR ANN and the corresponding output. 'A' and 'B' are the inputs of the OR gate and y is the output, which is binary. But the ANN deals with bipolar inputs and outputs.

Table 2: Input-Output Set

A	B	x_1	x_2	v_{dd}	v_{ss}	y_1	y_2	y_3	y_4	y_5	y_6	y'	y
0	0	-1	-1	1	-1	1	-1	-1	-1	-1	1	-1	0
0	1	-1	1	1	-1	1	-1	-1	-1	-1	-1	1	1
1	0	1	-1	1	-1	-1	-1	-1	-1	-1	-1	1	1
1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	1	1

Table 3 shows various weights used in the CMOS OR ANN.

Table 3: Weight Set

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}	w_{12}
1	-1	1	1	1	1	1	-1	1	1	1	1

V. Discussion

This paper is an approach to design the ANN model for basic logic gates like OR gate. A similar approach has already been done before by the same paper author but the ANN designed was deviated from the working of the CMOS gates. These ANNs are based on simple Self Organizing Maps (SOMs) which uses unsupervised learning. Therefore the synaptic weighs are fixed here. The weights are negative where inverted transistors are used. Other weights are kept positive. Work has already been done in this field as discussed in the

introduction but the work in this paper is an approach to design very simple circuits. Simple circuits are easy to implement. This approach can be used to simplify complex circuits in future for various applications.

Here, only OR-Gate is considered because the logic behind working of OR-Gate if mapped with CMOS OR-Gate, which is further mapped to a simple ANN, any type of logic circuit can be mapped using ANN. The concept of v_{dd} and v_{ss} is very clearly mentioned here. v_{dd} is always binary '1' as in CMOS OR-Gate and v_{ss} is always binary '0'. The function of inverter can easily be implemented with negative weight. Further, the emphasis is given on mapping ANN circuits to replace complicated electronic hardware circuits with ANN models which can be implemented using software.

VI. Conclusion

The approach in this paper is to design simple circuits to develop basic logic gates using CMOS. CMOS circuits are used in many devices now days. ANNs are also becoming popular now days. This approach will lead to the development of simple ANN circuits and also the hardware implementation of the ANN models will become simple with the advancement of this technology.

References

[1.] L. Fausett, "Fundamentals of Neural Networks, Architectures, Algorithms and Applications", Pearson Education, India, 2009.

[2.] G.N. Swamy, G. Vijay Kumar, "Neural Networks", Scitech, India, 2007.

[3.] Ashutosh Aggarwal, Rajneesh Rani, RenuDhir, "Handwritten Devanagri Character Recognition using Gradient Features", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 5, May 2012, ISSN 2277 128X, pp. 85-90.

[4.] Sandeep, Saha, Nabarag Paul, Sayam Kumar Das, Sandip Kundu, "Optical Character Recognition using 40-point Feature Extraction and Artificial Neural Network", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 4, April 2013, ISSN 2277 128X, pp. 495-502.

[5.] Ali Borji, Mandana Hamidi, Fariborz Mahmoudi, "Robust Handwritten Character Recognition with Features Inspired by Visual Ventral Stream", © Springer Science+Business Media, LLC. 2008, published online (31 August 2008), pp. 97-111.

[6.] Y Perwej and A Chaturvedi, “Neural Networks for Handwritten English Alphabet Recognition”, International Journal of Computer Applications, Volume 20, No. 7, pp. 1-5, 2011.

[7.] Frye R C, Rietman E A, and Wong C C, “Back-propagation learning and non idealities in analog neural network hardware,” Neural Networks, IEEE Transactions on, vol. 2, no. 1, pp. 110–117, 1991.

[8.] Jung S and Kim S S, “Hardware implementation of a real-time neural network controller with a dsp and an fpga for nonlinear systems,” Industrial Electronics, IEEE Transactions on, vol. 54, no. 1, pp. 265–271, 2007.

[9.] Hikawa H, “{FPGA} implementation of self organizing map with digital phase locked loops”, Neural Networks, vol. 18, no. 56, pp. 514 – 522, 2005, {IJCNN} 2005. Available Online: http://www.sciencedirect.com/science/article/pii/S0893608005001103

[10.] Merolla P A, Arthur J V, Alvarez-Icaza R, Cassidy A S, Sawada J, Akopyan F, Jackson B L, Imam N, Guo C, Nakamura Y, Brezzo B, Vo I, Esser S K, Appuswamy R, Taba B, Amir A, Flickner M D, Risk W P, Manohar R, and Modha D S, “A million spiking-neuron integrated circuit with a scalable communication network and interface”, Science, Vol. 345, No. 6197, pp 668–673, 2014.
Available Online: http://www.sciencemag.org/content/345/6197/668.abstract

[11.] Forssell M, “Hardware Implementation of Artificial Neural Networks”, 18-859E INFORMATION FLOW IN NETWORKS, pp 1-4, “Available: http://users.ece.cmu.edu/~pggrover/teaching/files/NeuromorphicComputing.pdf”, (Accessed : 2016)

[12.] Yellamraju S, Kumari Swati, Girolkar S, Chourasia S and Tete A D, “Design of Various Logic Gates in Neural Networks”, Annual IEEE India Conference (INDICON), 2013, Mumbai, India.

[13.] Hawas N M, Rekaby B K A, “ANN Based On Learning Rule Of Neuron Activation Function Using Electronic Devices”, International Journal of Advanced Computer Technology (IJACT), Vol 4, No. 3, pp 19-22, 2015.

[14.] Kale N B, Padole V B, “Compression and Decompression of Signal Using CMOS Technology...A Review”, International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 4, Issue 3, pp 53-55, 2014.

[15.] CMOS Gate Circuitry, Chapter 3 - Logic Gates,
Available: http://www.allaboutcircuits.com/textbook/digital/chpt-3/cmos-gate-circuitry, 2016

[16.] Hui W, Dejang C, Wei Z, Ping Z, Yongsheng L, “Application of artificial neural networks to segmented arc heather failure prediction”, AMSE Journals, Advances in Modeling, Series B, Vol. 54-1, pp 17-29. 2011