Multilayer Perceptron Analog Hardware Implementation Using Low Power Operational Transconductance Amplifier

Sherif Abden
Dept. of Electronics Engineering
German University in Cairo
Cairo, Egypt
sherif.abdullah@student.guc.edu.eg

Eman Azab
Dept. of Electronics Engineering
German University in Cairo
Cairo, Egypt
eman.azab@guc.edu.eg

2020 32nd International Conference on Microelectronics (ICM) | 978-1-7281-9664-0/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICM50269.2020.9331786

***Abstract*—This paper presents analog hardware implementation of multilayer perceptron (MLP) using operational transconductance amplifier (OTA) that is implemented and simulated in CMOS 180nm technology with 1.8V supply. The implemented circuits using the OTA perform addition, multiplication and activation which are the needed operations for any MLP. The components count in each circuit is small which allows implementing larger circuits. These circuits are current-mode (CM) circuits which makes the addition operation very straightforward and needs no power consumption using KCL. The power consumption and bandwidth of the OTA are 6.75μW and 32 KHz, respectively.**

***Index Terms*—neural networks, multilayer perceptron, analog hardware, activation functions, vector matrix multiplication, low power**

I. Introduction

In 1990, Carver Mead introduced the term neuromorphic computing for the first time in the literature [1]. The target was to implement neural networks (NNs) in hardware. One of the well-known types of NNs is the multilayer perceptron (MLP). MLPs were a popular machine learning solution in the 1980s, finding applications in diverse fields such as speech recognition, image recognition, and machine translation software [2]. Developers of early systems emphasized that it was possible to achieve much faster NN computations with custom chips and circuits [3]. The basic mathematical operations that an MLP does in order to calculate the output are: multiplication, addition and activation. To implement an MLP in hardware, the implemented circuit should perform these mathematical operations.

The addition operation can be easily implemented in current-mode circuits using KCL by summing all the current signals at a node. For the multiplication operation, many analog CM multipliers were implemented in the literature. Some of them are implemented for NN applications [4-6]. However, the circuits implemented in [4,5] had large number of transistors which makes them unsuitable for implementing larger MLPs. The circuit implemented in [6] had a relatively high power consumption and a bad linearity. Other analog CM multipliers were reported in literature [7-9]. However, their power consumption was relatively high. For the activation, the activation is defined as the process of calculating the output of a node in the MLP using nonlinear functions. There are many types of activation functions for MLPs. Activation functions like the sigmoid and hyperbolic tangent are popular in MLPs. However, they contain exponential terms which make their hardware implementations consumes large amount of power [10,11]. One of the well-known activation functions is the rectified linear activation function (ReLU) which is easy to implement in analog hardware.

In this paper, a low power OTA is implemented. This OTA is used as a building block for implementing the multiplier circuit and the ReLU activation function circuit. These two circuits can implement a complete perceptron which is the building unit of the MLP.

This paper is organized as follows. The mathematics behind the MLPs are described in section II. Section III represents the implemented analog hardware including the components of the multiplier circuit and the ReLU circuit. Section IV shows the simulation results of the complete perceptron and comparisons with other implementations from the literature. Finally, the paper is concluded in section V.

II. Multilayer Perceptron

The target of this section is to have a minimum level of knowledge about the MLPs and define the required mathematical equations before going into hardware implementation.

A. Perceptron Definition

A perceptron is a node in the MLP that does two mathematical operations to its inputs; summation and activation of the summation result. The inputs are weighted as each input is multiplied by a certain value which represents the effect of an input on the output, i.e., a change in an input with a small weight has a small effect on the output and vice versa. The mathematical model of a perceptron is given by the following equation:

$$y = f(\sum_{i=1}^{n} w_i.a_i + b) \quad (1)$$

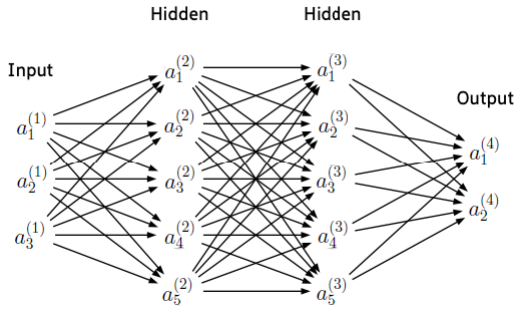


Fig. 1. An MLP with four layers of perceptrons

where a_i is the i^{th} input, w_i is the weight of the i^{th} input, b is a bias value, $n+1$ is the number of inputs including the bias value, and f is the activation function which is he ReLU activation function. It is a piecewise linear function that outputs the input directly if the input is positive, otherwise, the output is zero. It is represented by the following equation:

$$f(x) = max(0, x) = \begin{cases} x & \text{if } x \geq 0, \\ 0 & \text{if } x \leq 0. \end{cases} \quad (2)$$

B. MLP Definition

An MLP is a type of artificial NNs (ANNs). It consists of three types of perceptron layers; input layer, output layer, and hidden layers. It can have many hidden layers depending on the application. Fig. 1 shows an example of an MLP with four layers. Each perceprton is connected to all the perceptrons that precede and succeed it. However, the perceptrons in the same layer are not connected together and there is no feedback to the perceptron itself.

C. Forward Propagation

Forward propagation is the process of calculating the output of the MLP. Equation (1) illustrates the operation of a single perceptron. To include all perceptrons in one layer and to model the propagation from layer j to layer $j+1$, a vector-matrix multiplication is utilized as shown in the next equations:

$$\begin{bmatrix} a_0^{(j+1)} \\ a_1^{(j+1)} \\ . \\ a_K^{(j+1)} \end{bmatrix} = f(\begin{bmatrix} w_{0,0}^{(j)} & w_{0,1}^{(j)} & .. & w_{0,L}^{(j)} \\ w_{1,0}^{(j)} & w_{1,1}^{(j)} & .. & w_{1,L}^{(j)} \\ . & . & .. & . \\ w_{K,0}^{(j)} & w_{K,1}^{(j)} & .. & w_{K,L}^{(j)} \end{bmatrix} \begin{bmatrix} a_0^{(j)} \\ a_1^{(j)} \\ . \\ a_L^{(j)} \end{bmatrix}) \quad (3)$$

where K is the number of perceptrons in the $(j+1)^{th}$ layer, L is the number of perceptrons in the j^{th} layer, and $w_{n,m}^{(j)}$ is the weight value of the link from the the m^{th} perceptron in layer $j+1$ to the n^{th} perceptron in layer j. Bias values are represented using extra perceptron in each layer, that is a_0^x, where x is the layer number.

III. MLP Hardware

A. Vector-matrix Multiplier

Vector-matrix multiplier (VMM) is a CM circuit that performs the multiplication of the input vector and the weights

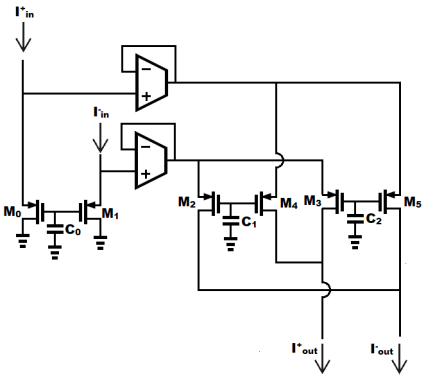
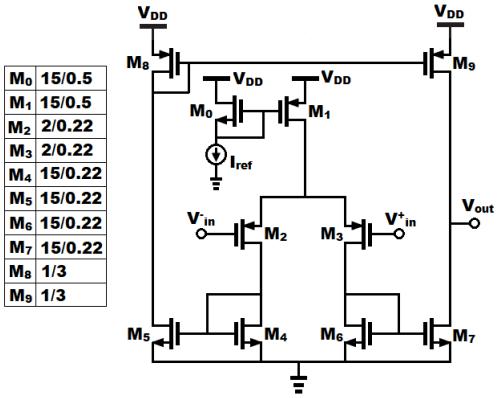


Fig. 2. 1×1 VMM schematic



M_0	15/0.5
M_1	15/0.5
M_2	2/0.22
M_3	2/0.22
M_4	15/0.22
M_5	15/0.22
M_6	15/0.22
M_7	15/0.22
M_8	1/3
M_9	1/3

Fig. 3. Operational transconductance amplifier schematic

matrix It performs one of the two main mathematical operations in forward propagation of MLPs. The VMM circuit is implemented using floating gate MOSFET-based current mirrors and operational transconductance amplifiers. The vector-matrix multiplication includes two operations; multiplication and addition. Since this is a CM circuit, the addition can be done easily using KCL by summing all the current signals at one node. For multiplication, a 1×1 VMM is implemented as shown in Fig. 2. A larger VMM can be implemented by adding more 1×1 VMMs together. To allow the full quadrant operation of the circuit, differential signals concept is utilized [12]. The input current, the weight and the output current are differential.

The 1×1 VMM is based on two circuits: an OTA and a Floating gate-based current mirror (FGCM). The implemented OTA is shown in Fig. 3 including transistors sizing in μm. The OTA is based on three current mirrors; (M_4,M_5), (M_6,M_7), and (M_8,M_9). M_2 and M_3 are the input transistors. The current mirror formed by M_0 and M_1 biases the input stage of the OTA. The reference current is set to 2.5μA. For the FGCM, it is a current mirror implemented using a MOSFET that has its gate connected to a capacitor. The capacitor electrically isolates the gate from any DC path. The charge of the capacitor controls the voltage of the gate. The idea of controlling the

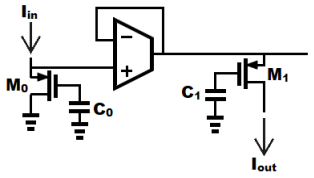


Fig. 4. Floating-gate MOSFET Current Mirror schematic

gate voltage allows the circuit to mirror the current with a scale without changing the dimensions of the MOSFETs. The FGCM circuit is shown in Fig. 4.

Assuming subthreshold operation of the transistor and $V_{sd} \gg 100mV$, the source to drain DC current is given by the following equation:

$$I_{sd} = I_0 \frac{W_p}{L_p} \exp\left(\frac{V_{sg} - |V_{th_p}|}{nV_t}\right) \tag{4}$$

where I_0 is the pre-exponential current, W_p and L_p are the width and length of the transistor, respectively, V_{th_p} is the threshold voltage, n is a technology dependant term, and V_t is the thermal voltage.

As a result of the negative feedback applied on the OTA shown in Fig. 4, the relation between source voltages of M_0 and M_1 is given by equation 5.

$$\frac{V_{s_1}}{V_{s_0}} = \frac{G_m}{G_m + g_{m_1}} \tag{5}$$

where G_m is the transconductance of the OTA, and g_{m_1} is the transconductance of M_1. From equation 5, the implemented buffer allow M_0 and M_1 to have approximately the same source voltage as long as $G_m \gg g_{m_1}$. However, the gate voltages can be changed via programming the capacitors C_0 and C_1. Thus, the following equation calculates the output current of the circuit:

$$I_{out} = I_{in} \exp\left(\frac{V_{C_1} - V_{C_0}}{nV_t}\right) \tag{6}$$

where V_{C_0} and V_{C_1} are the voltages applied to the gates of M_0 and M_1 due to the charge stored in C_0 and C_1, respectively. Using (6), the current signals I^+_{out} and I^-_{out} for the 1×1 VMM shown in Fig. 2 are given by (7) and (8), respectively:

$$I^+_{out} = I^+_{in} \exp\left(\frac{V_{C_1} - V_{C_0}}{nV_t}\right) + I^-_{in} \exp\left(\frac{V_{C_2} - V_{C_0}}{nV_t}\right) \tag{7}$$

$$I^-_{out} = I^+_{in} \exp\left(\frac{V_{C_2} - V_{C_0}}{nV_t}\right) + I^-_{in} \exp\left(\frac{V_{C_1} - V_{C_0}}{nV_t}\right) \tag{8}$$

By subtracting (8) from (7), the differential output current is given by the following equation:

$$I^+_{out} - I^-_{out} = (I^+_{in} - I^+_{in})(W^+ - W^-) \tag{9}$$

where

$$W^+ = \exp\left(\frac{V_{C_1} - V_{C_0}}{nV_t}\right), W^- = \exp\left(\frac{V_{C_2} - V_{C_0}}{nV_t}\right) \tag{10}$$

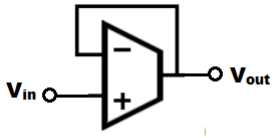


Fig. 5. ReLU activation function circuit schematic

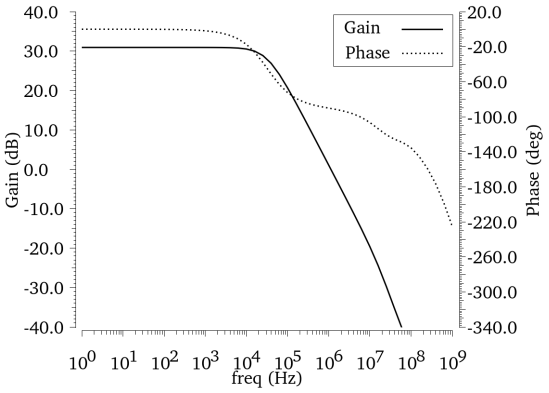


Fig. 6. Frequency response of the OTA

B. ReLU Activation Function Circuit

Hardware implementation of the ReLU function is very simple. Unlike sigmoid and hyperbolic tangent activation functions, the ReLU function does not contain exponential terms. Since the implemented OTA is biased with a positive supply and no negative supply exists, the OTA can be used to have a buffer that can implement the ReLU activation function. The ReLU activation function circuit schematic is shown in Fig. 5.

IV. Simulation Results

In this section, the circuits introduced in previous section are simulated using Cadence Virtuoso software with TSMC 180nm technology. Simulation results of the OTA, the 1×1 VMM, and the ReLU circuit are included.

Fig. 6 shows the frequency response of the OTA. The summation of all currents in the OTA branches is equal to 3.75μA, because all the current mirrors in the OTA mirror the current equally, which gives total power consumption of 6.75μW. To address the linearity of the OTA, the total harmonic distortion (THD) is measured when applying a ±50mV small signal input at 10KHz. The THD is calculated using the fast Fourier transform (FFT) tool in the simulator. The THD is equal to -38dB. Table II shows all these simulation results.

For the 1×1 VMM, the desired characteristics are linearity and power consumption. For the linearity, the maximum error in the ratio between the input and the output is found to be 3%. For the power consumption, the output current is much smaller than the current of the OTA circuit. So, the power consumption is mainly by the OTA. Each 1×1 VMM consumes 13.5μW because it has two OTAs. The DC characteristics of the circuit is shown in Fig. 7 for different weight values. The input range

TABLE I
OTA SIMULATION RESULTS

Parameter	Value
Transconductance	35 μS
Small signal voltage gain	31 dB
Phase Margin	89^o
3-dB bandwidth	32 KHz
Power Consumption	6.75 μW

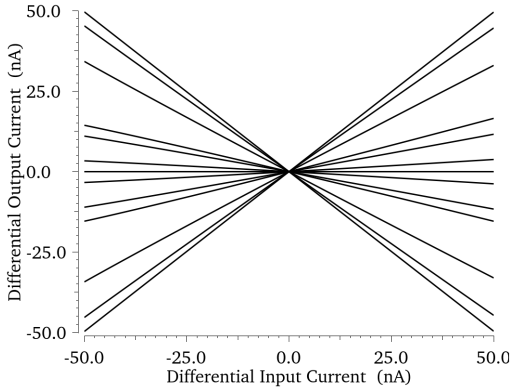


Fig. 7. DC characteristics of the 1×1 VMM for different weight values

is ±50nA.

For the ReLU circuit, it contains an OTA-based buffer. It consumes 6.75μW. From figure 8, it is obvious that for negative input, the buffer output is zero. For positive input, the buffer output is the same as the input. These voltage transfer characteristics are the same as the ReLU function.

Table II shows a comparison of the proposed multiplier and other works. Thee proposed multiplier has more moderate number of transistors. However, the power consumption is less than all the other multipliers. Even for low power circuits in [4,5], these two circuits had bad linearity and larger number of transistors. Regarding the activation function, the literature focuses in implementing other types of activation functions rather than the ReLU like the sigmoid and hyperbolic tangent [13]. The exact power consumption of activation functions implemented in [10,11] is not mentioned directly. However, form the available results by the latter works. they consume more power than the circuit implemented in this work.

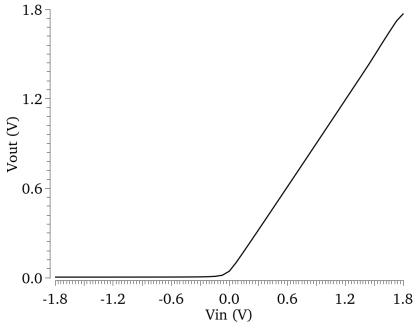


Fig. 8. DC characteristics of the ReLU activation function circuit

TABLE II
COMPARISON OF THE PROPOSED MULTIPLIER AND OTHER WORKS

Parameter	this work	[4]	[5]	[6]	[7]	[8]	[9]
Process (nm)	180	130	350	180	180	80	180
Supply (V)	1.8	1.2	1.2	1.2	1.2	1.8	2
Power (μW)	13.5	15	23	76.8	630	89.2	146.5
THD (dB)	-38	-32.5	-32	-27.5	-42.5	-40	-
Number of transistors	24	40	35	12	40	12	12

V. CONCLUSION

In this paper, a low power analog hardware implementation of a VMM and ReLU function circuit is presented. These two circuits are the main building blocks for implementing an MLP in hardware. The VMM consumes 13.5μW per one multiplication. The ReLU function circuit consumes 6.75μW. These circuits have small number of transistors which allows the integration in bigger systems. The circuits are implemented in CMOS 180nm technology.

REFERENCES

[1] C. Mead, "Neuromorphic electronic systems," Proceedings of the IEEE, vol. 78, no. 10, pp. 1629–1636, Oct 1990.
[2] P. D. Wasserman and T. Schwartz, "Neural networks. II. What are they and why is everybody so interested in them now?," in IEEE Expert, vol. 3, no. 1, pp. 10-15, Spring 1988.
[3] Schuman, Catherine Potok, Thomas Patton, Robert Birdwell, J. Dean, Mark Rose, Garrett Plank, James. (2017). A Survey of Neuromorphic Computing and Neural Networks in Hardware.
[4] F. M. Cardoso, M. C. Schneider and E. P. Santana, "CMOS analog multiplier with high rejection of power supply ripple," 2018 IEEE 9th Latin American Symposium on Circuits Systems (LASCAS), Puerto Vallarta, 2018, pp. 1-4.
[5] A. J. S. de Sousa et al., "A Very Compact CMOS Analog Multiplier for Application in CNN Synapses," 2019 IEEE 10th Latin American Symposium on Circuits Systems (LASCAS), Armenia, Colombia, 2019, pp. 241-244.
[6] D. Y. Aksin, P. B. Basyurt and H. U. Uyanik, "Single-ended input four-quadrant multiplier for analog neural networks," 2009 European Conference on Circuit Theory and Design, Antalya, 2009, pp. 307-310.
[7] I. Aloui, N. Hassen and K. Besbes, "Low-voltage low-power four-quadrant analog multiplier in current-mode," 2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Monastir, 2017, pp. 163-167.
[8] A. Baharmast, S. J. Azhari and S. Mowlavi, "A new current mode high speed four quadrant CMOS analog multiplier," 2016 24th Iranian Conference on Electrical Engineering (ICEE), Shiraz, 2016, pp. 1371-1376.
[9] A. Tijare and P. Dakhole, "CMOS current mode analog multiplier," 2016 International Conference on Signal and Information Processing (IConSIP), Vishnupuri, 2016, pp. 1-5.
[10] S. Azizian, K. Fathi, B. Mashoufi and F. Derogarian, "Implementation of a programmable neuron in 0.35m CMOS process for multi-layer ANN applications," 2011 IEEE EUROCON - International Conference on Computer as a Tool, Lisbon, 2011, pp. 1-4.
[11] G. Khodabandehloo, M. Mirhassani and M. Ahmadi, "Analog Implementation of a Novel Resistive-Type Sigmoidal Neuron," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 20, no. 4, pp. 750-754, April 2012.
[12] C. R. Schlottmann and P. E. Hasler, "A Highly Dense, Low Power, Programmable Analog Vector-Matrix Multiplier: The FPAA Implementation," in IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 1, no. 3, pp. 403-411, Sept. 2011.
[13] Schuman, Catherine Potok, Thomas Patton, Robert Birdwell, J. Dean, Mark Rose, Garrett Plank, James. (2017). A Survey of Neuromorphic Computing and Neural Networks in Hardware.