Neural Synaptic Weighting With a Pulse-Based Memristor Circuit

Hyongsuk Kim, *Member, IEEE*, Maheshwar Pd. Sah, Changju Yang, Tamás Roska, *Fellow, IEEE*, and Leon O. Chua, *Life Fellow, IEEE*

***Abstract*—A pulse-based programmable memristor circuit for implementing synaptic weights for artificial neural networks is proposed. In the memristor weighting circuit, both positive and negative multiplications are performed via a charge-dependent Ohm's law ($v = M(q) \times i$). The circuit is composed of five memristors with bridge-like connections and operates like an artificial synapse with pulse-based processing and adjustability. The sign switching pulses, weight setting pulses and synaptic processing pulses are applied through a shared input terminal. Simulations are done with both linear TiO_2 memristor and window-based nonlinear memristor models.**

***Index Terms*—Adjustability, memristor, programmable synapse, pulse-based processing, synaptic weight.**

I. Introduction

DESIGN of simple weighting circuits for synaptic multiplication between arbitrary input signals and weights is extremely important in artificial neural systems. There have been some research efforts to build neuron-like analog neural networks, in which on-board training as well as parallel processing capability are implemented [1]–[3]. Due to difficulty in implementing the synapses efficiently, this research has had only limited success so far. However, technologies to build analog neural networks for vision processing in which training is not necessary, but only parallel processing is essential, has nearly reached a commercial state. One active branch of this research area is the cellular neural networks (CNNs) [4]–[10], where massive synaptic multiplication circuits are implemented in a CMOS chip [7]. However, since shrinking the current transistor size is very difficult, introducing a more efficient approach is essential for further development of neural network implementations.

Recently, the Stanley Williams group has announced the successful development of a very compact memory element called memristor made of titanium dioxide [11], which exhibited the pinched hysteresis loop finger print. By applying a voltage or current to the memristor, its resistance can be altered in a manner similar to bio-neurons. The memristor was postulated by L. O. Chua as the fourth basic element of electrical circuits [12], [13]. It is based on a nonlinear relationship between charge and flux. Following the successful development of the Williams group, many researchers have worked actively on memristor models [14]–[19] and various applications of memristor devices [20]–[27]. Furthermore, Jo *et al.* recently reported the development of a new type of memristor which shows improved analog characteristics using Ag and Si in a sandwiched layer [25].

Many researchers in the field of memristors have suggested that this device has high potential for implementing artificial synapses [11], [28], [29]. In particular, Snider has presented a memristor-based self-organized network employing dedicated connections for inhibitory (negative) weighting [28].

Our proposed memristor synapse circuit is more general than Snider's in the sense that a full range (both positive and negative) of weights can be implemented. The pulses for setting the weight, and its sign (positive or negative) as well as the processing signals are applied to our proposed memristor circuit through a shared common input line.

The proposed memristor-based weighting circuit is composed of five memristors and one differential amplifier. The memristors provide a bridge-like switching for achieving either a positive or negative weighting. Though five memristors are employed to emulate a synapse in our proposed circuit, the total area of the memristors is less than that of a single transistor.

In this paper, the processing mechanism of the proposed memristor-based weighting is verified via simulations. Also, the weight setting and the synaptic weighting principle of our memristor-based synaptic weighting circuits are demonstrated.

In Section II, the principle and the mathematical model of the TiO_2 memristor, which is used as the fundamental model of our simulation, is introduced. The proposed memristor-based weighting circuit is described in Section III. In Section IV, the undesirable memristance variation caused by the synaptic processing pulse is analyzed. The simulation results and conclusion are given in Sections V and VI, respectively.

II. Memristor Models

The element constitutive relation of a memristor is defined by a nonlinear algebraic relation between its charge q and its flux φ by $\varphi = \hat{\varphi}(q)$.

Taking the derivative on both sides of $\varphi = \hat{\varphi}(q)$ leads to

$$\frac{d\varphi}{dt} = \frac{d\hat{\varphi}(q)}{dq} \cdot \frac{dq}{dt} \tag{1}$$

Manuscript received July 04, 2010; revised November 05, 2010; accepted June 28, 2011. Date of publication August 08, 2011; date of current version January 11, 2012. This work was supported in part by the US Air Force under Grant FA9550-10-1-0290 and the 2008 research funds of Chonbuk National University, Korea. This paper was recommended by Associate Editor M. Di Marco.

H. Kim, M. P. Sah, and C. Yang are with the Division of Electronics and Information Engineering, Chonbuk National University, Jeonju 561-756, Korea (e-mail: hskim@jbnu.ac.kr).

T. Roska is with the Computer and Automation Research Institute, Hungarian Academy of Sciences, Budapest H-1518, Hungary (e-mail: roska@sztaki.hu).

L. O. Chua is with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA (e-mail: chua@eecs.berkeley.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSI.2011.2161360

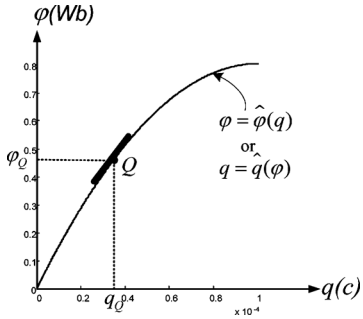


Fig. 1. A nonlinear function between flux φ and charge q in TiO_2 memristor [11]. Memristance $M(q)$ can be obtained by taking the slope at $q = q_Q$.

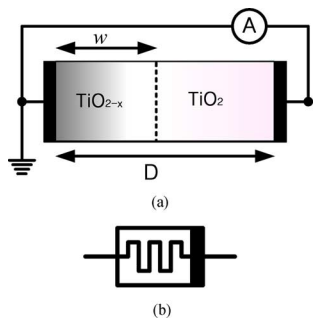


Fig. 2. (a) Structure of the TiO_2 memristor. TiO_2 and TiO_{2-x} layers are sandwiched between two platinum electrodes. When a voltage/current is applied, its memristance (resistance of the memristor in Ohms)/memductance (conductance of the memristor in siemens) is altered. (b) Symbol of the memristor.

which implies

$$v(t) = \frac{d\hat{\varphi}(q)}{dq} i(t) \equiv M(q)\ i(t) \tag{2}$$

where $M(q)$ is a charge-controlled resistance defined as

$$M(q) = \frac{d\hat{\varphi}(q)}{dq}. \tag{3}$$

According to (3), $M(q)$ is the slope at an operating point $q = q_Q$ on the $\varphi - q$ curve as in Fig. 1.

Since the resistance in this type of device is variable depending upon the operating point $q = q_Q$, and since $q = q_Q$ remains fixed when $v(t) = 0$ and $i(t) = 0$, the device can be used as a nonvolatile memory. Thus, the resistance $M(q)$ is called *memristance*.

Fig. 2(a) shows the structure of the TiO_2 memristor [11]. In the TiO_2 memristor, when voltage/current is applied to the device, the border line between the TiO_2 and TiO_{2-x} layers shifts with a function of the applied voltage/current. In consequence, the resistance between the two electrodes is altered. Fig. 2(b) shows the symbol of the memristor with the polarity indicated by a black bar at one end. The defined polarity indicates that for the $\varphi - q$ curve in Fig. 1(a), the memristance is reduced when current flows from the left side to the right (black bar) side.

Let the thickness of the TiO_{2-x} area be w, which is called a state variable. Let D be the thickness of the sandwiched area in the TiO_2 memristor, and let $R_{\rm ON}$ and $R_{\rm OFF}$ denote the low resistance and the high resistance values, respectively.

Then, the relation between the voltage and the current is given by

$$v(t) = \left(R_{\rm ON} \frac{w(t)}{D} + R_{\rm OFF} \left(1 - \frac{w(t)}{D} \right) \right) i(t). \tag{4}$$

The relationship between the state variable w and the current is defined by

$$\frac{dw(t)}{dt} = \mu_V \frac{R_{\rm ON}}{D} i(t) \tag{5}$$

where μ_v is the dopant mobility. This model is called a linear drift model since the velocity of the width is linearly proportional to the current.

Integration of (5) yields

$$w(t) = \mu_V \frac{R_{\rm ON}}{D} \int_0^t i(\tau) d\tau + w_0 \tag{6}$$

where w_o is the initial width of w.

It follows from (4) that the memristance $M(\mathrm{t})$ can be expressed by

$$M(t) = R_{\rm ON} \frac{w(t)}{D} + R_{\rm OFF} \left(1 - \frac{w(t)}{D} \right). \tag{7}$$

Plugging (6) into (7) yields

$$M(t) = \left\{ R_{\rm off} \left(1 - \frac{\mu_v R_{\rm ON}}{D^2} \left(1 - \frac{R_{\rm ON}}{R_{\rm OFF}} \right) \int_0^t i(\tau) d\tau \right) + \frac{w_0 R_{\rm OFF}}{D} \left(\frac{R_{\rm ON}}{R_{\rm OFF}} - 1 \right) \right\}. \tag{8}$$

Since the memristance is defined by the relationship between flux and charge as $M(t) = d\varphi(t)/dq(t)$, the integration of both sides of (8) yields

$$\varphi(t) = R_{\rm off} \left[q(t) \left(1 + \frac{w_0}{D} \left(\frac{R_{\rm ON}}{R_{\rm OFF}} - 1 \right) \right) - \frac{\mu_v R_{\rm ON}}{2D^2} \left(1 - \frac{R_{\rm ON}}{R_{\rm OFF}} \right) q(t)^2 \right] + \varphi_0. \tag{9}$$

Equation (9) indicates the flux-charge relationship shown in Fig. 1. Also, from (4) and (8), the relationship between the memristor voltage and the memristor current is given by

$$v(t) = \left\{ R_{\rm off} \left(1 - \frac{\mu_v R_{\rm ON}}{D^2} \left(1 - \frac{R_{\rm ON}}{R_{\rm OFF}} \right) \int_0^t i(\tau) d\tau \right) + \frac{w_0 R_{\rm OFF}}{D} \left(\frac{R_{\rm ON}}{R_{\rm OFF}} - 1 \right) \right\} i(t). \tag{10}$$

This relationship is the expression of a current-controlled memristor as in [15].

Differently from the linear drift model described above, nonlinear phenomena often appear at the boundaries of nanoscale devices; with even the small voltage applied across nanometer devices, a large electric field is produced and therefore, the ion boundary is moved in a nonlinear fashion in the nanoscale devices [16]. To show that the proposed memristor-based weighting circuit works also with the nonlinear model, the cases with nonlinear memristor models are investigated.

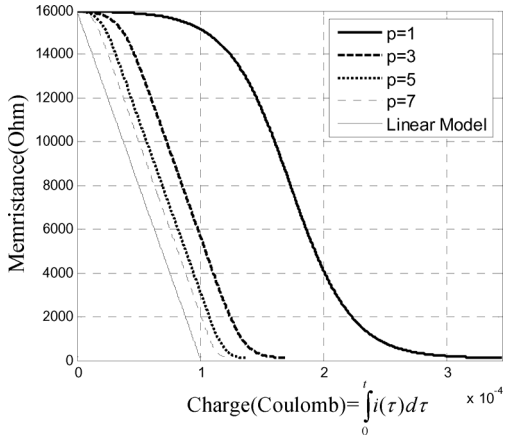


Fig. 3. Relationship between memristance versus charge for the linear and the nonlinear models [16] of memristors. As the integer p increases, the graphs tend to the linear model.

The nonlinear phenomenon can be modeled by multiplying the right side of (5) by a window $F_p(w)$, namely,

$$\frac{dw(t)}{dt} = \mu_V \frac{R_{\rm ON}}{D} i(t) F_p(w) \tag{11}$$

where p is a parameter with integer number. This is called the nonlinear drift model (or nonlinear model). Several different types of nonlinear memristor models with different window functions [16] have been proposed. One reasonable window function is defined by

$$F_p(w) = 1 - \left(2\frac{w}{D} - 1\right)^{2p}. \tag{12}$$

It is difficult to find the solution of $w(t)$ satisfying both (11) and (12) analytically. However, $w(t)$ can be computed by updating numerically as

$$w(t + \Delta T) \approx \mu_V \frac{R_{\rm ON}}{D} \left(1 - \left(2\frac{w(t)}{D} - 1\right)^{2p}\right) \Delta q + w(t) \tag{13}$$

where Δq is the charge increment fed to the memristor during the time interval ΔT and computed by integrating the input pulse $I(\mathrm{t})$ as

$$\Delta q = \int I(t)dt = I\Delta T. \tag{14}$$

From (7) and (13), the memristance can be expressed as a function of $q = \int_0^t i(\tau)d\tau$.

Fig. 3 shows the graphs of the memristance versus charge of the linear and nonlinear models of memristors. As the number p becomes smaller, the nonlinearity increases. On the contrary, as the integer p increases, the model tends to the linear model. At the middle regions of the curves, the increment rate of the memristance with respect to the charge, dM/dq, is constant for all the cases. This fact shows that the memristance can be calculated linearly at the middle region regardless of the value of p (degree of nonlinearity). Such linear characteristics in the middle region would be very useful in practical applications and could be investigated extensively.

The same parameters listed at the beginning of the simulation section (Section V) are used for generating these graphs.

III. Memristor Weighting Circuit

The proposed memristor-based weighting circuit is a novel and general circuit for implementing artificial synapses. The memristance (resistance of the memristor), which is a non-volatile memory, can be adjusted by changing the input current or voltage. Signal processing is performed with pulses and multiplication is implemented in a very simple way by using Ohm's law ($v = i \times M\,(q)$). The pulses for sign setting, weight setting, and processing signals share a common input line. Both negative and positive weights can be programmed.

Throughout this study, it is assumed that the voltage which is generated internally by the current signal makes the same amount of memristance change as that by the same magnitude of the externally applied voltage.

The proposed memristor neural weighting circuit is composed of five memristors and one set of differential amplifier as shown in Fig. 4(a). The memristor M_W at the center plays the weighting role of the synapse and the other memristors, Ms1, Ms2, Ms3, and Ms4, function as switches for selecting the weighting sign. The memristors alter the sign of the input voltage of the differential amplifier by switching the on-off states of the memristors in the manner of a bridge circuit. Though five memristors are needed in the proposed circuit, the total area of the five TiO_2 memristors is nevertheless smaller than that of a single transistor, because the minimum size of the memristor is one order of magnitude (two orders of magnitude in area) less than that of a transistor. The compatibility of the memristor with existing CMOS circuits has already been established in [28].

Fig. 4(b) shows a conventional architecture of artificial synapses with a simplified Gilbert multiplier that is employed in the CNN structure [7]. Recalling that the total size of five memristors is smaller than that of a transistor, the size benefit of the memristor-based weighting circuit is obvious. Another difference of the conventional synapse in [7] from the proposed memristor-based weighting circuit is that the terminals providing the input signal and weighting value in conventional circuits are separated, while in the proposed memristor synapses, pulses for sign setting, weight setting (biasing), and multiplication processing are all applied through a single input terminal.

A. Sign Setting

Prior to weight setting or multiplication (weighting) processing, the sign of the synaptic weight is set first. This is implemented by applying a strong current pulse (high amplitude or wide current pulse) to change the memristances toward one of the two extreme values, turn-on and turn-off resistances. (In the TiO_2 memristor [11], we assume a turn-on resistance of 116 Ω and turn-off resistance of 15.98 KΩ.)

Let us apply a wide positive or negative current pulse at the input terminals of Fig. 5(a) and (b), respectively. The current at each memristor is computed as described in the Appendix. The current at each memristor is then integrated to compute its incremental charge. Then, the charge at each memristor is updated and its memristance is computed.

When a positive wide pulse is applied at the input terminal of the memristor circuit, as shown in Fig. 5(a), the charge of both

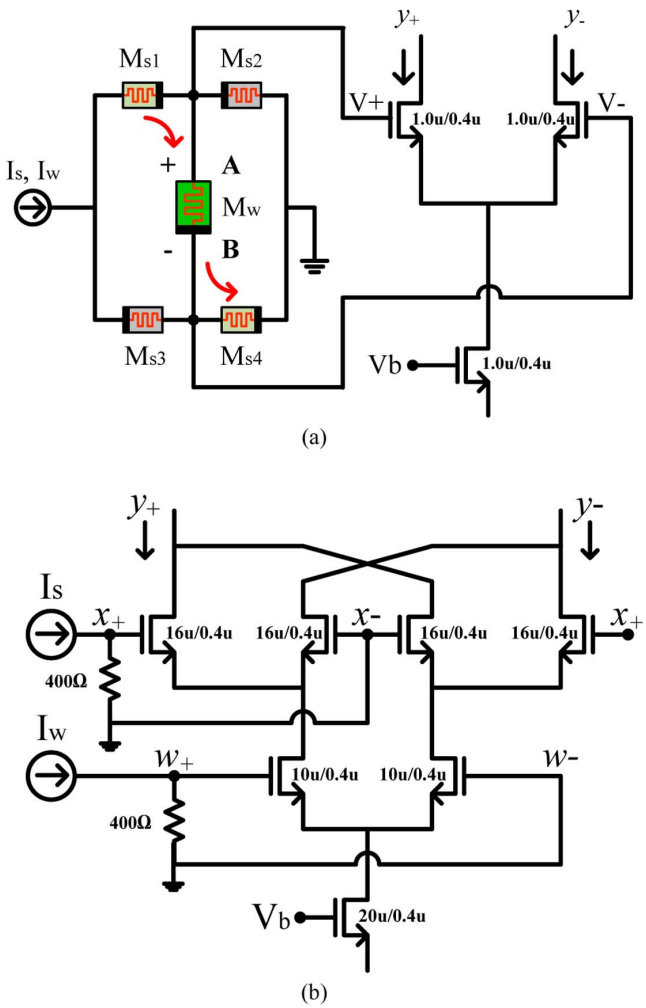


Fig. 4. Artificial synaptic circuits. (a) Memristor-based synaptic circuit. (b) Conventional CMOS synaptic circuit [7].

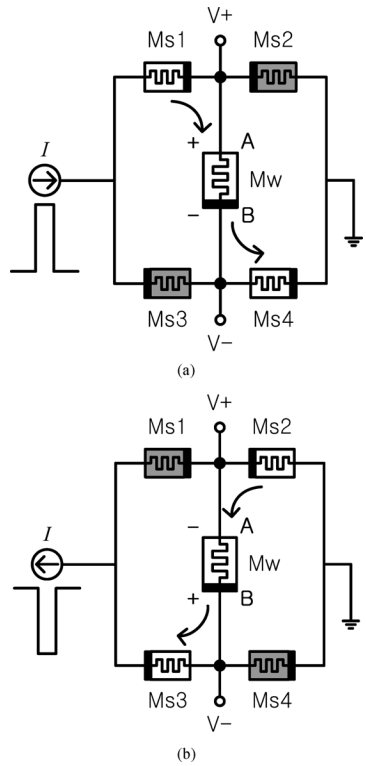


Fig. 5. Configurations of the memristor-based sign switching circuit. (a) Positive weighting configuration (Ms1 = Ms4 = Min., Ms2 = Ms3 = Max.). (b) Negative weighting configuration (Ms1 = Ms4 = Max., Ms2 = Ms3 = Min.).

memristors Ms1 and Ms4 increases while that of Ms2 and Ms3 decreases. Note that the polarities of memristors Ms1 and Ms4 are opposite from those of Ms2 and Ms3. Accordingly, the magnitude of the resistances of memristors Ms1 and Ms4 decreases, while that of Ms2 and Ms3 increases. If the magnitude or the duration of the pulse is large enough, the memristances of memristors Ms1 and Ms4 will reach their minimum values, while those of Ms2 and Ms3 will attain their maximum values, thereby resulting in on (low memristance) state for Ms1 and Ms4, and off (high memristance) state for Ms2 and Ms3. Afterward, any current signal appearing at the input terminal passes mostly through Ms1, M_W, and Ms4. Consequently, a larger voltage is produced at terminal V+ than that at terminal V− with a positive current i. In this case, memristor M_W functions as a positive weight thereafter.

On the contrary, when a negative wide pulse is applied as shown in Fig. 5(b), memristors Ms2 and Ms3 will be switched on and memristors Ms1 and Ms4 will be switched off. Afterward, any current signal appearing at the input terminal passes mostly through Ms2, M_W and Ms3. Consequently, the voltage appearing at terminals V+ and V− will be opposite to that of the configuration shown in Fig. 5(a). In this case, the value M_W functions as a negative weight thereafter.

TABLE I
WEIGHTING-SIGN SWITCHING IN THE MEMRISTOR-BASED SYNAPSE OF FIG. 5
(Min. = 116 Ω, Max. = 15.98 KΩ)

Applied Pulse	Configuration	Input Current	(V+ - V-)	Sign of Voltage at Mw
Positive Big/Wide Pulse	Ms1=Ms4=Min.	+	+	+
	Ms2=Ms3=Max.	−	−	
Negative Big/Wide Pulse	Ms1=Ms4=Max.	+	−	−
	Ms2=Ms3=Min.	−	+	

In the case of TiO_2 memristor, a rectangular current pulse with amplitude of 10 mA and width of about 20 ms is sufficient for sign switching. A summary of the weight-sign switching process in the proposed memristor-based weighting circuit is shown in Table I. When a positive pulse (resp., negative) is applied, our memristor-based synapse will always result in a positive (resp., negative) weighting.

B. Weight Setting

After the sign-setting operation, we proceed to set the memristance for the synaptic weight. A smaller pulse with an appropriate sign (in the range [0, 0.7 ms]) is applied for this purpose until the memristance of Mw reaches to its desirable weighting value. The equivalent range of the memristance M_W is [116 Ω, 1 KΩ] in the TiO_2 memristor model.

C. Synaptic Weight (Multiplication) Processing

Synaptic multiplication can be implemented after setting M_W with the appropriate weight. The synaptic weighting processing is the multiplication of the current by the memristance via Ohm's law, $v = i \times M$, which is implemented by simply letting the current flow through the memristor.

When the input signal pulse i_s appears at the input terminal, most of the current flows through the memristors in the on-state in Fig. 4(a). Let the voltage across M_w be $v_{\rm diff}$, then

$$v_{\rm diff} = i_s \times M_w. \tag{15}$$

Accordingly, the differential current at the output of the differential amplifier is

$$i_{\rm out} = g_m v_{\rm diff} = g_m i_s \times M_w \tag{16}$$

where g_m is the transconductance of the transistor pair in the differential amplifier.

Another component of the current appearing at the output of the differential amplifier is the common-mode current created from memristors Ms2 or Ms4. The common-mode output current $i_{\rm CM}$ is given by

$$i_{\rm CM} = \frac{g_m \times v_{\rm cm}}{1 + g_m r_o} \tag{17}$$

where $v_{\rm CM}$ is the common voltage appearing across the small memristances Ms2 and Ms4. Let this memristance be M_G. The memristance M_G is normally the same as M_W since weight setting is implemented after initializing all the memristors with a large/wide sign-setting pulse.

The common-mode current component (17) becomes

$$i_{\rm CM} = \frac{g_m i_s \times M_G}{1 + g_m r_o}. \tag{18}$$

Therefore, the output current i_O is given by

$$i_o = i_{\rm diff} + i_{\rm CM} = g_m i_s \times M_w + \frac{g_m i_s \times M_G}{1 + g_m r_o}. \tag{19}$$

Since memristors M_W and M_G are normally programmed to have the same memristance, as discussed above, the corresponding output current will be

$$i_O = g_m' i_s \times M_w \tag{20}$$

where g_m' is

$$g_m' = g_m + \frac{g_m}{1 + g_m r_o}. \tag{21}$$

As a result, as long as M_G is programmed to have the same memristance as M_w, the output current i_o is proportional to the product of i_s and M_w. Therefore, the memristor circuit in Fig. 4(a) functions as an analog synaptic weighting circuit.

D. Effects Upon Input Noise

The effects of our memristor circuit on noise depend upon the locations where noises are presented.

In the case that some noise appears at the input terminal in Fig. 4(a), the noise and the input signal are not separable and the noise is treated as a part of the input signal.

Let the input noise current signal be i_n. Then, the total input current with noise i_s' is

$$i_s{}' = i_s + i_n \tag{22}$$

where the signal-to-noise ratio S/N at the input terminal is $({\rm S/N})_{\rm in} = i_s/i_n$.

Using (20), the output current i_o' after synaptic weighting is given by

$$i_o' = g_m' i_s' \times M_W = g_m' i_s \times M_W + g_m' i_n \times M_w = i_{\rm os} + i_{\rm on} \tag{23}$$

where $i_{\rm os}$ and $i_{\rm on}$ denote the signal and noise components of the output current, respectively. Therefore, the S/N value at the output of the circuit is $i_{\rm os}/i_{\rm on} = i_s/i_n$, which is identical to that at the input terminal. This shows that the noises are neither amplified nor suppressed in the proposed circuit.

On the other hand, in the case that noise appears on the connection lines between the memristors and the differential amplifiers, they are all suppressed due to the common-mode rejection feature of differential amplifiers.

IV. Memristance Variation and Doublet Input

A. Memristance Drifting

One possible problem of the memristor-based processing is the memristance (weight value) drifting due to flux accumulation of input signal pulses. In this section, the memristance variation resulting from synaptic processing (weighting) pulses is analyzed. In contrast to the sign setting and weight programming pulses which have a fixed amplitude with varying pulse widths, the synaptic processing (weighting) pulses have varying amplitudes with a fixed width.

Utilizing the current $I_i(t)$ at each memristor, which is derived in the Appendix, the charge Δq_i fed to the memristor during time interval ΔT is computed using (14).

In the linear case (without window), the derivative of (8) with respect to q gives

$$\frac{dM_i}{dq} = R_{\rm off} \frac{\mu_v R_{\rm ON}}{D^2} \left(\frac{R_{\rm ON}}{R_{\rm OFF}} - 1 \right). \tag{24}$$

Therefore, the memristance change ΔM_i resulting from the charge change Δq_i can be obtained from (24) as

$$\Delta M_i \approx R_{\rm off} \frac{\mu_v R_{\rm ON}}{D^2} \left(\frac{R_{\rm ON}}{R_{\rm OFF}} - 1 \right) \Delta q_i. \tag{25}$$

Equation (25) is the memristance change of memristor i resulting from input $I({\rm t})$ during time interval ΔT.

In the nonlinear case, the expression of the memristance change is more complicated. From (13), the variation of the doped area Δw_i caused by Δq_i of the $i_{\rm th}$ memristor in Fig. 3(a) is given by

$$\begin{aligned} \Delta w_i(t) &\approx w_i(t + \Delta T) - w_i(t) \\ &= \mu_V \frac{R_{\rm ON}}{D} \left(1 - \left(2\frac{w(t)}{D} - 1 \right)^{2p} \right) \Delta q_i. \end{aligned} \tag{26}$$

It follows from (7) that the change of the memristance $\Delta M_i(t)$ is given by

$$\Delta M_i(t) \simeq M_i(t+\Delta T) - M_i(t) = (R_{\rm ON} - R_{\rm OFF})\,\frac{\Delta w_i(t)}{D}. \quad (27)$$

Substituting (26) into (27), we obtain

$$\Delta M_i(t) \approx (R_{\rm ON} - R_{\rm OFF}) \times \mu_V \frac{R_{\rm ON}}{D^2}\left(1 - \left(2\frac{w_i(t)}{D} - 1\right)^{2p}\right)\Delta q_i \quad (28)$$

where $w_i(t)$ is computed numerically by updating using (13). Observe from (27) that the memristance change in the nonlinear case is a nonlinear function of the normalized doped area $w_i(t)/D$.

B. Switch-Based Doublet Generator

As discussed before, the unipolar pulses will result in undesirable drift of the operating point. However, if the input pulse is a doublet with zero net area, then no change in memristance occurs.

In this subsection, we propose a high-speed doublet generator of doublet pulse with equal positive and negative areas. The doublet generator is composed of four switches as shown in Fig. 6(a), where two pairs of switches ($S1$ and $S2$, $S3$ and $S4$) operate concurrently. Fig. 6(b) shows the timing diagram. In the doublet generator, two switch pairs are controlled by the switching pulses PA and PB as shown in Fig. 6(b). During each half period $T_d = (1/2)T_s$, the switches $S1$ and $S2$ of one pair are turned on concurrently with PA, while those of the from other pair ($S3$ and $S4$) are turned off with PB. With this switch configuration, the input current signal is fed from node A to node B when the input signal is positive. Soon after the first pulse period, switches $S3$ and $S4$ are turned on for another time period T_d while switches $S1$ and $S2$ are off. During this time period, the same input signal is fed from node B to node A, thereby creating the opposite polarity from the previous one. As a result, a positive doublet $V_{\rm CD}$ is produced, as shown in Fig. 6(b).

If the input signal is negative, the current is fed from the opposite direction and a negative doublet, whose first sub-pulse is negative while the second is positive, is generated.

V. Simulations

Simulations were performed based on the TiO_2 memristor model [11]. The weight-sign setting and weight setting simulations were carried out on our memristor circuit using HSPICE (BSIM3v3.1, Level 49). The parameters used for these simulations are $R_{\rm ON} = 100\ \Omega$, $R_{\rm OFF} = 16\ {\rm K}\Omega$, $D = 10$ nm, $\mu_v = 10^{-14}\ {\rm m^2V^{-1}S^{-1}}$, w_0 (high resistance state) $= 0.001D$ and w_0 (low resistance state) $= 0.999D$. The parameter p of the window function of the nonlinear model [16] was 4.

Processings were performed by applying three different kinds of rectangular current pulses for weight-sign setting, weight setting, and synaptic weightings, respectively, where the pulse for the weight-sign setting is widest, while those for synaptic weightings are narrowest. The amplitudes of the pulses for weight-sign setting and synaptic weight setting are fixed at 10 mA, while their widths are variable. In contrast, the pulse amplitudes for the synaptic processing are sampled (variable) from input signals and their widths are fixed at 3 ns.

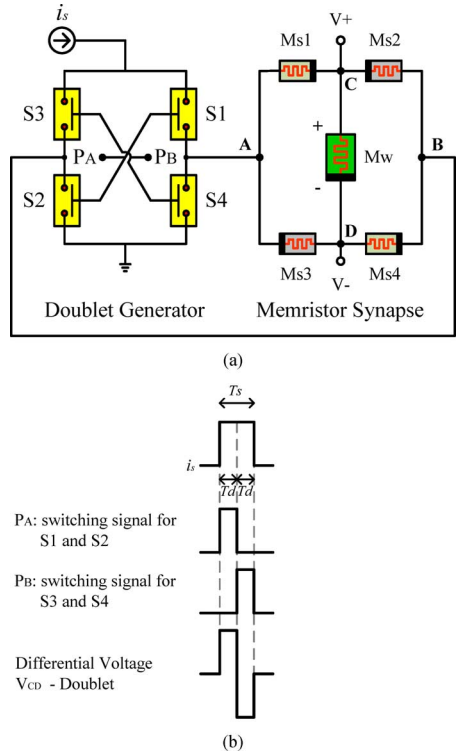


Fig. 6. (a) A switch-based doublet generator with a memristor weighting circuit. (b) Pulses appearing at each terminal in Fig. 6(a).

A. Weight-Sign Setting

Our memristor synaptic circuit allows positive and negative weighting. We assume that the initial memristances ${\rm Ms1} = {\rm Ms4}$ and ${\rm Ms2} = {\rm Ms3}$ are 15.98 KΩ and 116 Ω, respectively. Let the input current be I. From computation given in the Appendix, the voltage across M_W is $-0.98M_w \times I$. Therefore, the initial sign of the weighting with the above memristances is negative. Fig. 7(a) shows the transition of the memristance at each memristor while a positive current pulse with $+10$ mA amplitude is applied. At the end of the pulse, the memristances are turned into ${\rm Ms1} = {\rm Ms4} = 116\ \Omega$ and ${\rm Ms2} = {\rm Ms3} = 15.98\ {\rm K}\Omega$. Fig. 7(b) shows the variation of the current at each memristor. Since the output of the circuit is the voltage taken from M_W, the sign of the current I_W indicates the sign of weighting with this circuit. In Fig. 7(b), the current I_W which begins with $-0.98I$ ends with $+0.98I$, while a positive input pulse is applied. This implies that the sign of the weighting is switched from negative to positive. Note that both negative and positive signs can be obtained in the whole regions of the left and right sides of Fig. 7, respectively. However, a shorter time is taken to program the same amount of memristance value in the two extreme state regions than in the middle regions.

Similar to the positive sign transition as above, the negative sign can be implemented when a negative pulse is applied at a positive weighting state.

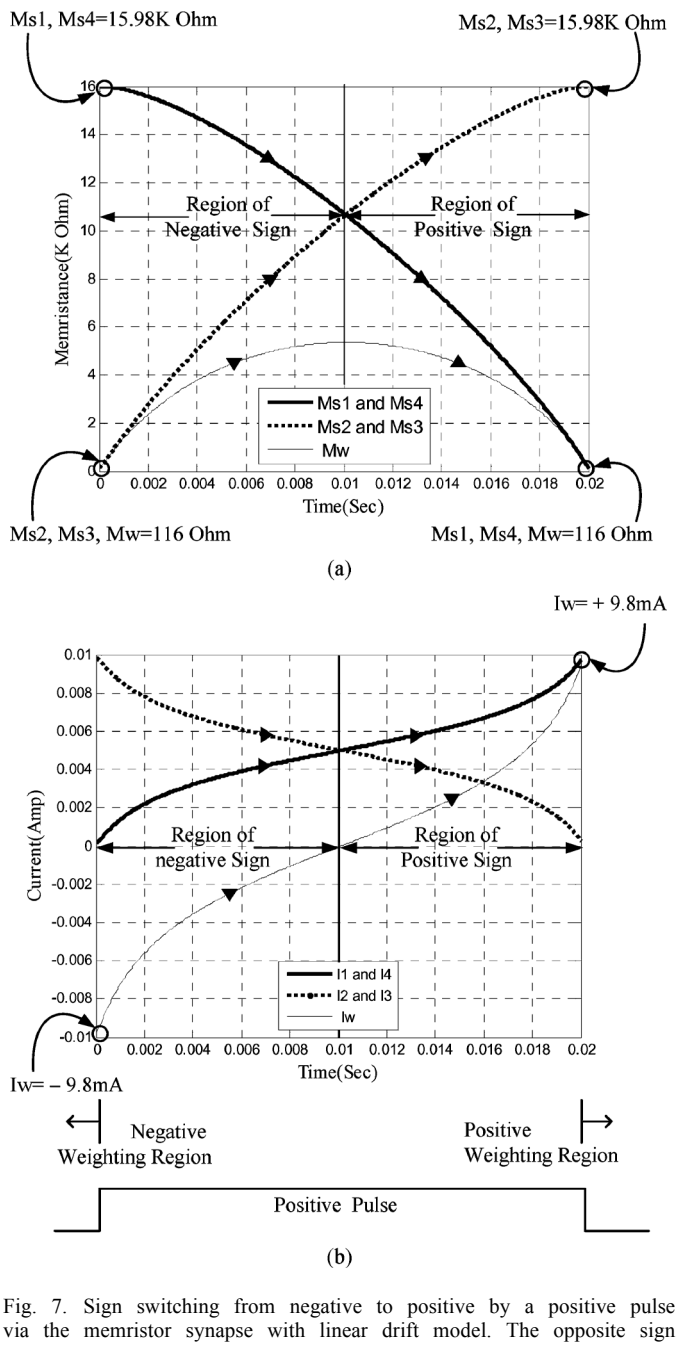


Fig. 7. Sign switching from negative to positive by a positive pulse via the memristor synapse with linear drift model. The opposite sign switching can be implemented by a negative pulse. (a) Memristance transition from $\{\mathrm{Ms1} = \mathrm{Ms4} = 15.98\ \mathrm{K}\Omega, \mathrm{Ms2} = \mathrm{Ms3} = 116\ \Omega\}$ to $\{\mathrm{Ms1} = \mathrm{Ms4} = 116\ \Omega, \mathrm{Ms2} = \mathrm{Ms3} = 15.98\ \mathrm{K}\Omega\}$. (b) Current variation at each memristor.

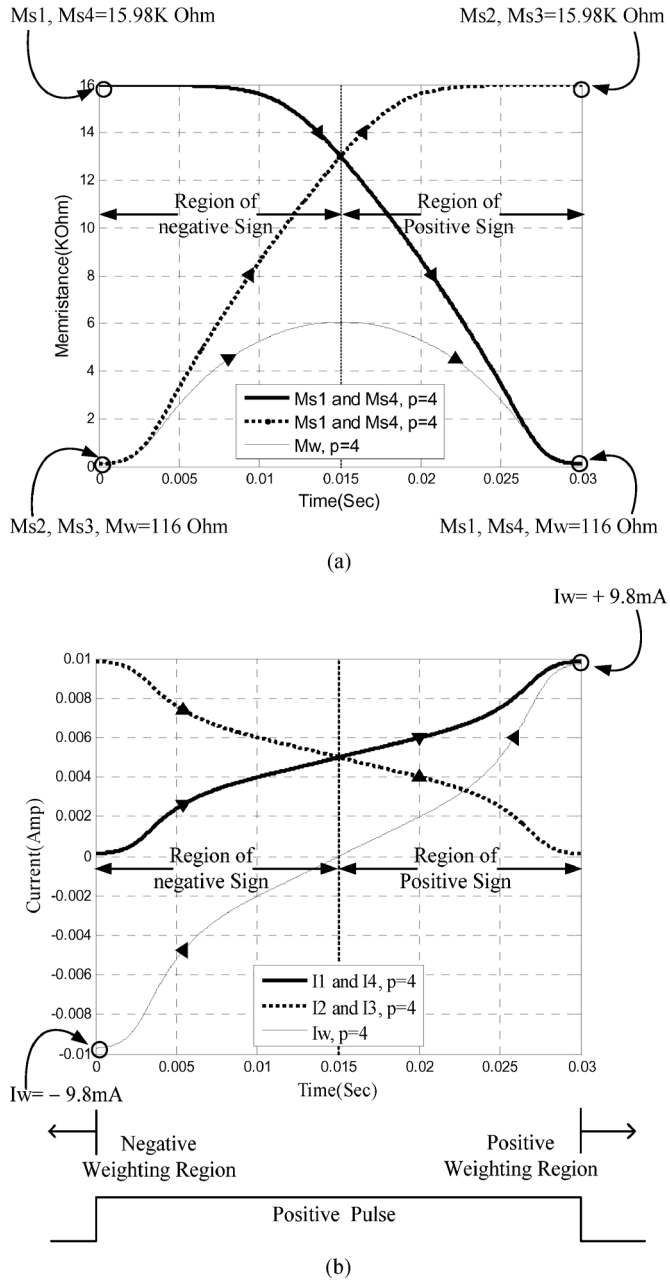


Fig. 8. Sign switching with the nonlinear drift model with the window function in [16] ($p = 4$). (a) Memristance transition from $\{\mathrm{Ms1} = \mathrm{Ms4} = 15.98\ \mathrm{K}\Omega, \mathrm{Ms2} = \mathrm{Ms3} = 116\ \Omega\}$ to $\{\mathrm{Ms1} = \mathrm{Ms4} = 116\ \Omega, \mathrm{Ms2} = \mathrm{Ms3} = 15.98\ \mathrm{K}\Omega\}$. (b) Current variation at each memristor.

The effect of the window function ($p = 4$) of the nonlinear model in [16] was also investigated via simulation. Fig. 8(a) and (b) shows the memristance transition and the corresponding current transitions, respectively. The memristance transition behavior is similar to that of the linear model; memristances are changing linearly in the middle region in both linear and nonlinear cases. Such linear characteristics in the middle region would be very useful in practical applications.

In the nonlinear case, the time (pulse width) for sign switching or weight programming takes longer than that with the linear model. Also, the memristance changing rate is very low at both extreme states (close to Max or Min state) while that at the middle region is relatively high.

B. Weight Setting

Once the sign setting of the weight is completed, the magnitude setting for the weight is conducted. The weight setting starts from the minimum level of M_w. A pulse with appropriated sign and width is applied at the input so that M_w becomes a desired value. The memristance range used for weight value setting is from 116 Ω to 1 KΩ, which can be generated by a pulse widths ranging over [0, 0.7 ms]. Fig. 9 shows the relationship between the memristance and the current pulse width for

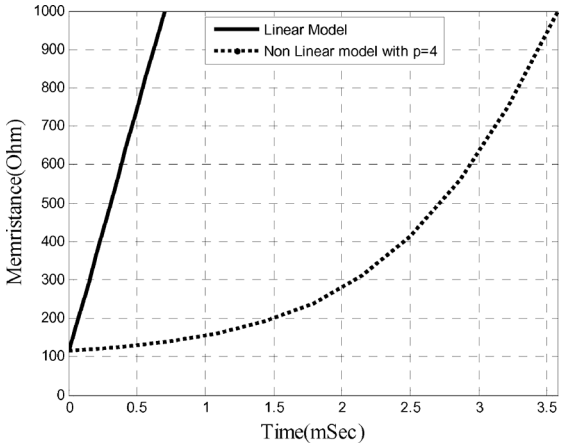


Fig. 9. Relationship between the memristance and the pulse width of the input current for linear and nonlinear memristor models.

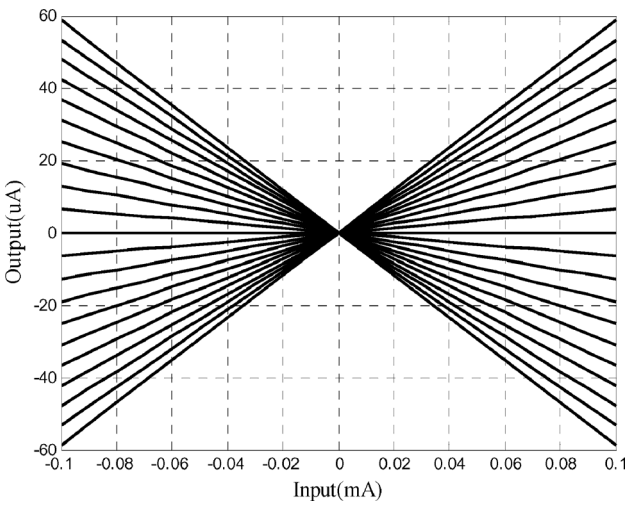


Fig. 10. Overall weighting (multiplication) performance of the memristor-based synaptic circuit. The ranges of multiplicands are $M_W = [-1\ \mathrm{K\Omega},\ 1\ \mathrm{K\Omega}]$ and $\mathrm{Input} = [\,0.1\ \mathrm{mA},\ 0.1\ \mathrm{mA}]$.

both linear and nonlinear models. As shown in Fig. 9, the memristance M_W of the linear model is nearly proportional to the applied pulse width while that with the nonlinear model is nonlinear. A nonlinear mapping between the desired memristance and the pulse width is needed for the weight programming of the nonlinear model, while the weight of the linear model is linearly programmable.

C. Synaptic Multiplication (Weighting)

The synaptic processing in artificial neural networks consists of the multiplication (weighting) between the input and the memristance. To evaluate the weighting (multiplication) performance of the proposed memristor-based synaptic circuit, extensive simulations were carried out. Fig. 10 shows the output of the differential amplifier when the range of M_W is $[-1\ \mathrm{K\Omega},\ 1\ \mathrm{K\Omega}]$ and that of the input is $[-0.1\ \mathrm{mA},\ 0.1\ \mathrm{mA}]$. As shown in the figure, the circuit shows excellent weighting (multiplication) performance compared to that of existing CMOS weighting circuit [7] in terms of linearity.

Let us recall the possible memristance (weight value) drifting problem due to input pulse accumulation during the synaptic multiplication period. The doublet form of inputs can resolve the problem since sub-pulses with opposite polarities compensate for changes in memristances. Difference between the positive and negative regions of the doublet should be as small as possible to reduce the memristance changing effect.

TABLE II
MEMRISTANCE CHANGES CAUSED FROM A SINGLET PULSE (1 mA, 3 ns) LINEAR CASE WITHOUT WINDOW FUNCTION

		$\Delta M_1,\Delta M_4\ (\Omega)$	$\Delta M_2,\Delta M_3\ (\Omega)$	$\Delta M_w\ (\Omega)$
Case 1	Ms1,Ms4 = 116 Ω Ms2,Ms3 = 15.98 kΩ Mw = 116 Ω	4.70×10^{-4}	6.80×10^{-6}	4.63×10^{-4}
Case 2	Ms1,Ms4 = 1 kΩ Ms2,Ms3 = 15.98 kΩ Mw = 1 kΩ	4.27×10^{-4}	5.03×10^{-5}	3.76×10^{-4}

TABLE III
MEMRISTANCE CHANGES CAUSED FROM A SINGLET PULSE (1 mA, 3 ns) NONLINEAR CASE WITH A WINDOW FUNCTION OF (11), $p = 4$

		ΔM1,ΔM4 (Ω)	ΔM2,ΔM3 (Ω)	ΔMw (Ω)
Case 1	Ms1,Ms4 = 116 Ω Ms2,Ms3 = 15.98 kΩ Mw = 116 Ω	7.47×10^{-6}	1.08×10^{-7}	7.36×10^{-6}
Case 2	Ms1,Ms4 = 1 k Ω Ms2,Ms3 = 15.98 kΩ Mw = 1 kΩ	2.63×10^{-4}	8.0×10^{-7}	2.32×10^{-4}

TABLE IV
MEMRISTANCE CHANGES CAUSED FROM A DOUBLET PULSE (1 mA, 3 ns) LINEAR CASE WITHOUT WINDOW FUNCTION

		ΔM1,ΔM4 (Ω)	ΔM2,ΔM3 (Ω)	ΔMw (Ω)
Case 1	Ms1,Ms4 = 116 Ω Ms2,Ms3 = 15.98 kΩ Mw = 116 Ω	5.64×10^{-8}	8.16×10^{-10}	5.56×10^{-8}
Case 2	Ms1,Ms4 = 1 kΩ Ms2,Ms3 = 15.98 kΩ Mw = 1 kΩ	5.12×10^{-8}	6.03×10^{-9}	4.51×10^{-8}

When a singlet pulse $p(\mathrm{t})$ with 1 mA amplitude and 3 ns width is applied to the memristor circuit, the memristance changes calculated from (25) and (28) are shown in Tables II and III for linear and nonlinear cases, respectively.

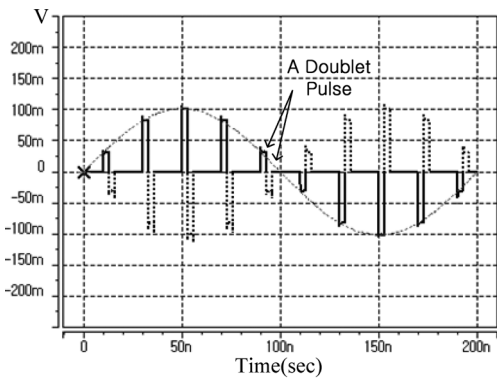
When the doublet input generated by the doublet generator in Fig. 6(a) is applied to the memristor circuit, the memristance changes are much smaller than those of the linear counterparts as shown in Tables IV and V for linear and nonlinear cases, respectively. This fact suggests that the doublet circuit lessens the memristance drifting problem during the weighting stage significantly.

However, for long time processing, readjustment of the weights would be necessary from time to time during idling periods.

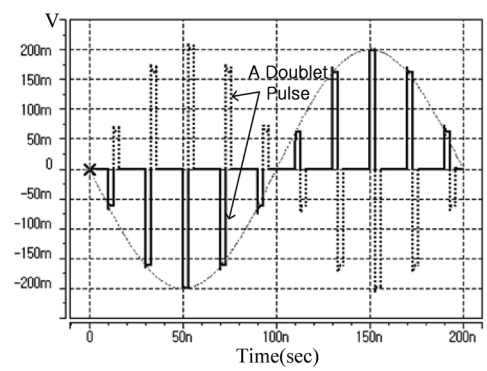
Fig. 11 illustrates the synaptic multiplication (weighting) with our memristor-based weighting circuit for $M_{W1} = 116\ \Omega$ set with a positive sign, and $M_{W2} = 232\ \Omega$ set with a negative

TABLE V
MEMRISTANCE CHANGES CAUSED FROM A DOUBLET PULSE (1 mA, 3 ns) NONLINEAR CASE WITH A WINDOW FUNCTION OF (11), $p = 4$

		$\Delta M_1, \Delta M_4\,(\Omega)$	$\Delta M_2, \Delta M_3\,(\Omega)$	$\Delta M_w\,(\Omega)$
Case 1	$Ms_1, Ms_4 = 116\ \Omega$ $Ms_2, Ms_3 = 15.98\ k\Omega$ $Mw = 116\ \Omega$	8.96×10^{-10}	1.3×10^{-11}	8.83×10^{-10}
Case 2	$Ms_1, Ms_4 = 1\ k\ \Omega$ $Ms_2, Ms_3 = 15.98\ k\Omega$ $Mw = 1\ k\Omega$	3.16×10^{-8}	9.6×10^{-11}	2.78×10^{-8}



(a)



(b)

Fig. 11. Illustration of synaptic weighting for two cases (a) $M_{W1} = 116\ \Omega$ with a positive sign and (b) $M_{W2} = 232\ \Omega$ which is two times the value of M_{W1} with a negative sign. The magnitude of the pulse train in the case of (b) is two times bigger than that of (a) and their signs are opposite to each other, as expected.

sign. The input pulses were sampled from a sinusoidal current signal and converted to doublets. The outputs are voltage differences taken at the input terminals of the differential amplifier. Fig. 11 shows that the output pulses with $M_{W2} = 232\,\Omega$ is two times larger than the case with $M_{W1} = 116\ \Omega$, as expected. The use of doublet signals is aimed for preventing the memristances from the unwanted drifting. For the subsequent processing with the ordinary MOS circuit, each doublet signal is converted to a signlet pulse by sampling the output signal at every first pulse time period. Note that the doublet pulses are synchronized with the external clock.

To summarize the processing of the proposed memristor-based synaptic weighting circuit, Fig. 12 illustrates a timing diagram of a successive procedure of sign switching, weight setting and synaptic processings. For the sign switching,

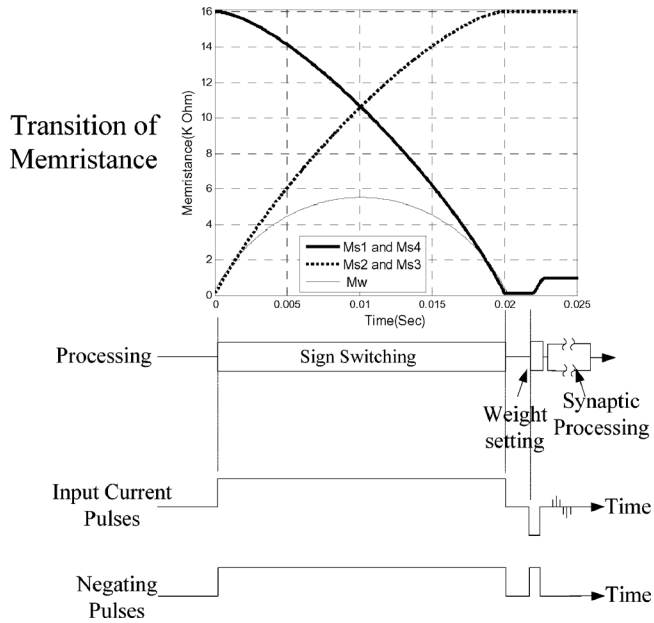


Fig. 12. Timing diagram of a successive procedure of sign switching, weight setting and synaptic processing.

a wide current pulse of 20 ms width and 10 mA magnitude is applied as shown at the bottom of Fig. 12. During the weight setting stage, a middle sized pulse ranged of [0, 0.7] ms is applied depending upon the desired magnitude of a weight. Note that at the synaptic processing (multiplication) stage, the doublet generators are enabled. Then, the output of each synaptic circuit is sampled during the first pulse period of the doublet pulse and utilized for further processing. Such synaptic processing is repeated until the next sign or weight setting command is issued.

VI. CONCLUSION

A memristor circuit to implement an artificial synapse is proposed. With the proposed memristor-based weighting circuit, the memristance (resistance of the memristor), which is a non volatile memory, can be adjusted by changing the input current or voltage.

Simulations were carried out via the *hp* TiO_2 memristor model. Weight-sign setting and weight setting simulations were carried out on the memristor circuits and the synaptic weighting simulation was carried via HSPICE. The weight-sign switching was implemented with a rectangular current pulse having an amplitude of 10 mA and width of 20 ms. The weight range of [116 Ω, 1 KΩ] is set by applying a pulse with a range of [0, 0.7 ms]. Over the range of the input $= [-0.1\ \mathrm{mA}, 0.1\ \mathrm{mA}]$ and $M_W = [-1\ \mathrm{K}\Omega, 1\ \mathrm{K}\Omega]$, the memristor weighting circuit exhibits excellent linearity both positive and negative weightings.

To avoid the possible change of memristances due to charge accumulation of input signal pulses, a switching-based doublet voltage generator is presented.

One of the unique features of our memristor synapse model is its pulse-based operation. The required control and processing are performed with input pulses. Since the weighting values of our memristor synapses could be programmed by input pulses,

it is expected to be applicable to neural network learning and sequential processing operations in cellular neural networks.

APPENDIX

Let $I(t)$ be an input current pulse in Fig. 5 and let $I_1(t)$, $I_2(t)$, $I_3(t)$, $I_4(t)$, and $I_w(t)$ be the current pulses at memristors M_1, M_2, M_3, M_4 and M_w, respectively. Then, Kirchoff's current law gives

$$I(t) = I_1(t) + I_3(t) \tag{A1}$$

$$I_2(t) = I_1(t) - I_w(t) \tag{A2}$$

$$I_3(t) = I_4(t) - I_w(t). \tag{A3}$$

Also, we have the voltage relationships

$$M_W I_W(t) + M_{S4} I_4(t) = M_{S2} I_2(t) \tag{A4}$$

and

$$M_{S2} I_2(t) + M_{S1} I_1(t) = M_{S4} I_4(t) + M_{S3} I_3(t). \tag{A5}$$

From the above five independent equations, we can derive the five current pulses $I_1(t)$, $I_2(t)$, $I_3(t)$, $I_4(t)$ and $I_w(t)$ (see equations (A6)–(A10) at the bottom of the page) as functions of the input current $I(t)$.

REFERENCES

[1] M. Holler, S. Tam, H. Castro, and R. Benson, "An electrically trainable artificial neural network (ETANN) with 10240 'Floating gate' synapses," *Artificial Neural Networks: Electronic Implementations, IEEE Computer Society Neural Networks Technology Series*, pp. 50–55, 1990.

[2] H. Withagen, "Implementing backpropagation with analog hardware," in *Proc. IEEE ICNN-94*, Orlando, FL, 1994, pp. 2015–2017.

[3] S. Lindsey and T. Lindblad, "Survey of neural network hardware invited paper," in *Proc. SPIE Application and Science of Artificial Neural Networks Conf.*, Apr. 1995, vol. 2492, pp. 1194–1205.

[4] L. O. Chua and L. Yang, "Cellular neural networks: Theory," *IEEE Trans. Circuits Syst.*, vol. CAS-35, no. 10, pp. 1257–1272, Oct. 1988.

[5] L. O. Chua and L. Yang, "Cellular neural networks: Applications," *IEEE Trans. Circuits Syst.*, vol. CAS-35, no. 10, pp. 1273–1290, Oct. 1988.

[6] T. Roska and L. O. Chua, "The CNN universal machine: An analogic array computer," *IEEE Trans. Circuits Syst. II*, vol. 40, no. 3, pp. 163–172, Mar. 1993.

[7] R. Domíýnguez-Castro, S. Espejo, A. Rodríguez-Vázquez, R. A. Carmona, P. Földesy, Á. Zarándy, P. Szolgay, T. Szirányi, and T. Roska, "A 0.8-μm CMOS two-dimensional programmable mixed-signal focal-plane array processor with on-chip binary imaging and instructions storage," *IEEE J. Solid-State Circuits*, vol. 32, no. 7, pp. 1013–1026, Jul. 1997.

[8] C.-D. Zheng, H. Zhang, and Z. Wang, "Improved robust stability criteria for delayed cellular neural networks via the LMI approach," *IEEE Trans. Circuits Syst. II, Expr. Briefs*, vol. 57, no. 1, pp. 41–45, Jan. 2010.

[9] H. Kim, T. Roska, H. Son, and I. Petras, "Analog addition/subtraction on the CNN-UM chip with short-time superimposition of input signals," *IEEE Trans. Circuits Syst. I*, vol. 50, no. 3, pp. 429–432, Mar. 2003.

[10] H. Kim, H. Son, T. Roska, and L. O. Chua, "High-performance viterbi decoder with circularly connected 2-D CNN unilateral cell array," *IEEE Trans. Circuits Syst. I*, vol. 52, no. 10, pp. 2208–2218, Oct. 2005.

[11] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, 2008.

$$I_1(t) = \left(\frac{M_{S3}M_W + M_{S3}M_{S2} + M_{S3}M_{S4} + M_{S4}M_W}{M_{S1}M_W + M_{S1}M_{S2} + M_{S1}M_{S4} + M_{S2}M_W + M_{S3}M_W + M_{S3}M_{S2} + M_{S3}M_{S4} + M_{S4}M_W} \right) I(t) \tag{A6}$$

$$\begin{aligned} I_2(t) = {} & \frac{M_W}{M_W + M_{S2} + M_{S4}} \\ & \times \left(\frac{M_{S3}M_W + M_{S3}M_{S2} + M_{S3}M_{S4} + M_{S4}M_W}{M_{S1}M_W + M_{S1}M_{S2} + M_{S1}M_{S4} + M_{S2}M_W + M_{S3}M_W + M_{S3}M_{S2} + M_{S3}M_{S4} + M_{S4}M_W} \right) I(t) \\ & + \frac{M_{S4} I(t)}{M_W + M_{S2} + M_{S4}} \end{aligned} \tag{A7}$$

$$I_3(t) = \left(1 - \frac{M_{S3}M_W + M_{S3}M_{S2} + M_{S3}M_{S4} + M_{S4}M_W}{M_{S1}M_W + M_{S1}M_{S2} + M_{S1}M_{S4} + M_{S2}M_W + M_{S3}M_W + M_{S3}M_{S2} + M_{S3}M_{S4} + M_{S4}M_W} \right) I(t) \tag{A8}$$

$$\begin{aligned} I_4(t) = {} & \frac{(M_W + M_{S2}) I(t)}{M_W + M_{S2} + M_{S4}} - \frac{M_W}{M_W + M_{S2} + M_{S4}} \\ & \times \left(\frac{M_{S3}M_W + M_{S3}M_{S2} + M_{S3}M_{S4} + M_{S4}M_W}{M_{S1}M_W + M_{S1}M_{S2} + M_{S1}M_{S4} + M_{S2}M_W + M_{S3}M_W + M_{S3}M_{S2} + M_{S3}M_{S4} + M_{S4}M_W} \right) I(t) \end{aligned} \tag{A9}$$

$$\begin{aligned} I_W(t) = {} & \frac{(M_{S2} + M_{S4})}{M_W + M_{S2} + M_{S4}} \\ & \times \left(\frac{M_{S3}M_W + M_{S3}M_{S2} + M_{S3}M_{S4} + M_{S4}M_W}{M_{S1}M_W + M_{S1}M_{S2} + M_{S1}M_{S4} + M_{S2}M_W + M_{S3}M_W + M_{S3}M_{S2} + M_{S3}M_{S4} + M_{S4}M_W} \right) I(t) \\ & - \frac{M_{S4} I(t)}{M_W + M_{S2} + M_{S4}} \end{aligned} \tag{A10}$$

[12] L. O. Chua, "Memristor—the missing circuit element," *IEEE Trans. Circuit Theory*, vol. CT-18, no. 5, pp. 507–519, Sep. 1971.
[13] L. O. Chua and S. M. Kang, "Memristive devices and systems," *Proc. IEEE*, vol. 64, no. 2, pp. 209–223, Feb. 1976.
[14] S. Shin, K. Kim, and S. M. Kang, "Compact models for memristors based on charge-flux constitutive relationships," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 29, no. 4, pp. 590–598, Apr. 2010.
[15] E. M. Drakakis, S. N. Yaliraki, and M. Barahona, "Memristors and Bernoulli dynamics," in *Proc. 2010 Int. Workshop on Cellular Nanoscale Networks and their Applications (CNNA)*, Feb. 2010, pp. 70–75.
[16] Y. Joglekar and S. Wolf, "The elusive memristor: Properties of basic electrical circuits," *Eur. J. Phys.*, vol. 30, pp. 661–675, 2009.
[17] B. O. Kavehei, A. Iqbal, Y. S. Kim, K. Eshraghian, S. F. Al-sarawi, and D. Abboti, "The fourth element: Characteristics, modeling and electromagnetic theory of the memristor," 2010, arXiv: 1002.3210v1 [cond-mat.mes-hall].
[18] F. Y. Wang, "Memristor for introductory physics," 2008, arXiv: 0808. 0286v1 [physics. Class-ph].
[19] M. Itoh and L. O. Chua, "Memristor cellular automata and memristor discrete-time cellular neural networks," *Int. J. Bifurcation and Chaos (IJBC)*, vol. 19, no. 11, pp. 3605–3656, 2009.
[20] A. Rak and G. Cserey, "Macromodelling of the memristor in SPICE," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 29, no. 4, pp. 632–636, Apr. 2010.
[21] H. Kim, M. P. Sah, Y. Changju, and L. O. Chua, "Memristor-based multilevel memory," in *Proc. 2010 Int. Workshop on Cellular Nanoscale Networks and their Applications (CNNA)*, Feb. 2010, pp. 89–94.
[22] Q. Xia, W. Robinett, M. W. Cumbie, N. Banerjee, T. J. Cardinali, J. J. Yang, W. Wu, X. Li, W. M. Tong, D. B. Strukov, G. S. Snider, G. Medeiros-Ribeiro, and R. S. Williams, "Memristor-CMOS hybrid integrated circuits for reconfigurable logic," *Nano Lett.*, vol. 9, no. 10, pp. 3640–3645, Sep. 2009, DOI: 10.1021/nl901874j.
[23] J. Borghettil, G. S. Snider, P. J. Kuekes, J. J. Yang, D. R. Stewart, and R. S. Williams, "Memristive switches enable stateful logic operations via material implication," *Nature Lett.*, vol. 464, no. 8, pp. 873–876, Apr. 2010.
[24] Y. V. Pershin and M. D. Ventra, "Practical approach to programmable analog circuits with memristors," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 8, pp. 1857–1864, Aug. 2010.
[25] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Lett.*, pp. 1297–1301, 2010, DOI: 10.1021/nl904092h.
[26] I. Petras, "Fractional-order memristor-based Chua's circuit," *IEEE Trans. Circuits Syst. II, Expr. Briefs*, vol. 57, no. 12, pp. 975–979, Dec. 2010.
[27] R. Riaza, "Nondegeneracy conditions for active memristive circuits," *IEEE Trans. Circuits Syst. II, Expr. Briefs*, vol. 57, no. 3, pp. 223–227, Mar. 2010.
[28] G. Snider, "Self-organized computation with unreliable, memristive nanodevices," *Nanotechnology*, vol. 18, no. 36, pp. 1–13, 2007.
[29] Y. Ho, G. M. Huang, and P. Li, "Dynamical properties and design analysis for nonvolatile memristor memories," *IEEE Trans. Circuits Syst. I*, vol. 58, no. 4, pp. 724–736, Oct. 2010.

Hyongsuk Kim (M'09) received the B.S. degree in electronics engineering from the Hanyang University, Korea, in 1980, the M.S. degree in electrical engineering from Chonbuk National University, Korea, in 1982, and the Ph.D. degree in electrical engineering from the University of Missouri, Columbia, in 1992.

During 1982–1993, he worked for the Agency for Defense Development, Korea. Since 1993, he has been a Professor with the Division of Electronics Engineering, Chonbuk National University, Korea. From 2000 to 2002 and again from 2009 to 2010, he was with the Nonlinear Electronics Laboratory, EECS Department, University of California, Berkeley, as a Visiting Scholar. His current research interests include the memristor and its application to cellular neural/nonlinear networks. He is also interested in VLSI chip design for analog parallel processing.

Maheshwar Pd. Sah received the B.E. degree from Nepal Engineering College, Nepal, in 2005, and the M.E. degree from Chonbuk National University, Korea, in 2010, where he is currently working toward the Ph.D. degree in electronics and information engineering.

His main research interests include circuit design, cellular neural networks, design of error correction circuits, analog Viterbi decoders, and analysis of memristors and memristive systems.

Changju Yang received the B.S. and M.S. degrees in electronics and information engineering in 2008 and 2010, respectively, from Chonbuk National University, Korea, where he is currently studying toward the Ph.D. degree in electronics and information engineering.

His main research interests include circuit design, analog Viterbi decoders, and analysis of memristors and memristive systems.

Tamás Roska (M'87–SM'90–F'93) received the Diploma in electrical engineering from the Technical University of Budapest, Hungary, in 1964, and the Ph.D. and D.Sc. degrees in Hungary in 1973 and 1982, respectively.

Since 1982, he has been with the Computer and Automation Research Institute of the Hungarian Academy of Sciences, where he is presently head of the Spatial-Temporal Computing Architectures Laboratory and Chair of the Scientific Council. He is also a Professor and was the founder Dean of the Faculty of Information Technology at the Pázmány P. Catholic University, Budapest, Hungary. In 1974 and since 1989 in each year until 2010, he was a Visiting Scholar at the University of California at Berkeley. His research interests are cellular wave computing, info-bionics, cellular neural networks, nonlinear circuits and systems, analogic spatial-temporal supercomputing and computational complexity. He has published more than 100 research papers and four books (some as a coauthor), receiving over 2000 citations for four seminal papers.

Prof. Roska was the founder Chair of the Technical Committee on Cellular Neural Networks and Array Computing in the IEEE Circuits and Systems Society. He is a Fellow of the IEEE and elected member of four Academies of Sciences in Europe.

Leon O. Chua (S'60–M'62–SM'70–F'74–LF'02) received the M.S. degree from the Massachusetts Institute of Technology, Cambridge, MA, in 1961, and the Ph.D. degree from the University of Illinois at Urbana-Champaign in 1964.

He became an Assistant Professor of Electrical Engineering at Purdue University in 1964, and was promoted to Associate Professor in 1967. He joined the University of California at Berkeley in 1970, where he is a Professor of electrical engineering and computer sciences. His research interests are in memristors, chaos, cellular automata, and cellular neural networks.

Dr. Chua is the first recipient of the 2005 Gustav Kirchhoff Award, the highest IEEE Technical Field Award for outstanding contributions to the fundamentals of any aspect of electronic circuits and systems that has a long-term significance or impact. He was also awarded the prestigious IEEE Neural Networks Pioneer Award in 2000 for his contributions in neural networks, the Guggenheim Fellow award in 2010, and a Leverhulme Trust Visiting Professorship in 2011.