VLSI Implementation of ART1 Memories

Suan W. Tsay and Robert W. Newcomb, *Fellow, IEEE*

***Abstract*—This paper presents a hardware implementation of long-term memory and short-term memory for binary input adaptive resonance theory (ART1) neural networks. This implementation is based on chemical–electrical interactions in real neurons which are known to control axon release of chemical materials which in turn modulate the conductances of synapses. An "axon-synapse-tree" structure is introduced to realize bottom-up long-term memory. The axon-synapse tree is realized by voltage modulation of synapse conductances.**

I. Introduction

THE ART (adaptive resonance theory) neural network, introduced by G. Carpenter and S. Grossberg [1], [2], is a self-organizing neural network which can learn unexpected input patterns very quickly and stably. As such it is a versatile neural network worthy of hardware realization in CMOS VLSI form. As a first step toward such realization, we consider here the memory structures of ART1, the binary-input version of ART.

Complex describing equations of the memories make the hardware implementation of ART memories difficult. To get around these difficulties, in this paper we use chemical–electrical interaction concepts to develop VLSI circuits to realize the different functions of ART memories. In so doing, we avoid some of the mathematically complex ART equations but we obtain equivalent behavior. Only binary-input ART systems are implemented at this time. This is because in realizing the MOS resistor used here we rely on the constraint that an MOS transistor is turned on only when the gate-to-source voltage is greater than the threshold voltage. Consequently, the ART mentioned in this paper will in all cases be ART1. However, in contrast to the binary input signals, all internal signals will be analog in nature.

In Section II we review the basic structure of ART neural networks. In section III we describe the ART long-term memories giving a VLSI realization. The main idea behind bottom-up long-term memory is Weber's law. Section III also includes a discussion of Weber's law and our means of approximating it in realizable hardware. The means of making shunting short-term memory, needed to prevent instability, is presented in Section IV.

II. Basic Structure of an ART Neural Network

An ART neural network [1], [2] is mainly composed of two sets of neurons, the top field F_T and the bottom field F_B, a gain control, and a reset subsystem as illustrated in Fig. 1. F_T and F_B contain a total of n of neurons, $N_1 \cdots N_n$, of state n-vector $\boldsymbol{x}$ with components $x_1, \cdots, x_n$, partitioned into the bottom state $\boldsymbol{x}_B$ composed of $x_1, \cdots, x_k$ and the top state $\boldsymbol{x}_T$ of $x_{k+1}, \cdots, x_n$. The state vector $\boldsymbol{x}$ forms the short-term memory (STM) (there being two portions, STM_T and STM_B). The gating strengths, that is, weights, on the bottom-up and top-down paths between the F_B and F_T fields form the long-term memory (LTM) (there again being two portions, LTM_{BU} and LTM_{TD}). These gating strengths are referred to here as synapse weights.

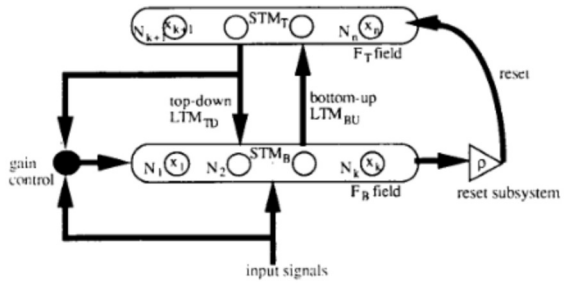


Fig. 1. Basic structure of an ART neural network. The input signal is the vector $\boldsymbol{I} = (I_1, I_2, \cdots, I_k)^T$, with T denoting transpose, I_i, $1 \le i \le k$, takes binary values; i.e., $I_i \in \{0, 1\}$.

The bottom (F_B field) neurons N_j, $1 \le j \le k$, take the input signal pattern k-vector $\boldsymbol{I}$ from the outside world and use it, along with gain control and weighted top-down signals, to send signals $f_j(x_j)$ through the bottom-up LTM to the F_T field. A top neuron N_m, $k+1 \le m \le n$, in the F_T field receives signals from all the bottom neurons in F_B and competes with its neighboring neurons in F_T to find the top neuron which receives the largest bottom-up signal. The result of the competition is that this distinguished F_T neuron, N_i, is activated to a "high" (that is, x_i takes on a large or high value) while all the other top neurons are suppressed to a "low." Then both LTM's associated with N_i begin to change their synapse weights. The bottom-up LTM synapse weights will grow or decay to some value according to Weber's law [1], [2]. The top-down LTM will code the pattern which shows up in the F_B field (that is, the top-down LTM's synapse weights will grow or decay to "high" or "low" according to the stable state of the F_B STM). The gain control subsystem, indicated as a solid circle in Fig. 1, is used to compare the input pattern $\boldsymbol{I}$ and the top-down LTM synapse strengths associated with the active F_T neuron. If the scalar-valued similarity between $\boldsymbol{I}$ and the top-down LTM is lower than the parameter ρ, which means the previously coded pattern in this top-down LTM does not match the input pattern $\boldsymbol{I}$, the reset subsystem will issue a reset signal to inhibit the active F_T neuron. ART will select another F_T neuron and compare its top-down LTM synapse strengths with the input pattern $\boldsymbol{I}$. This process continues until the "similarity" is greater than ρ. The LTM then begins to learn the new pattern, which is the signal represented in the STM_B (i.e., in the value of the k-vector states $\boldsymbol{x}_B$). Although concise, the above is a brief discussion of the basic operation of ART memories.

Manuscript received August 24, 1990; revised November 19, 1990. This work was supported by the ONR under Grant N00014-90-J1114.

The authors are with the Microsystems Laboratory, Electrical Engineering Department, University of Maryland, College Park, MD 20742.

IEEE Log Number 9042025.

1045-9227/91/0300-0214$01.00 © 1991 IEEE

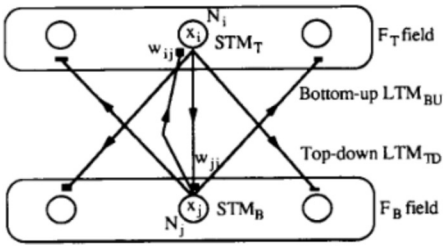


Fig. 2. ART memories. STM is the state of neurons, which is represented via circles in the F_B and F_T fields. The value of LTM is represented by the size of the solid boxes on the ends of the bottom-up and top-down signal paths, denoted w_{ij} from neuron j to neuron i.

As seen above, the memories of ART can be grouped into two categories, LTM in synapse weights and STM in neuron states, each of which has its specific describing equation. In order to discuss these we show a rough drawing of ART memories in Fig. 2.

The two LTM's perform different functions and thus play different roles in learning. Bottom-up LTM is used to gate the bottom-up signals so that the ART can decide which neuron in F_T will be activated to learn the signal. Top-down LTM, however, is used to code the pattern in the F_B field if the corresponding F_T neuron is activated after competition. STM is the "state" of all the neurons and its value changes depending on the excitatory and inhibitory input strengths. All the bottom-up and top-down LTM signals are excitatory to the destination neurons. The inhibitory signals for F_B neurons come from the gain control subsystem, while for the F_T neurons inhibition comes from the reset subsystem and neighboring F_T neurons.

III. Synapse Gated Long Term Memory

LTM is to be considered to represent gating strengths in biological synapses. The bottom-up LTM and top-down LTM synapse weights can increase or decrease only when their corresponding F_T neuron is active [3]. Consequently, the form of changes to w_{ij} depends upon whether w_{ij} is either for top-down connections, in which case its active F_T neuron is N_j, $k+1 \le j \le n$, or for bottom-up connections, in which case its active F_T neuron is N_i, $k+1 \le i < k$. Following [3, p. 23] we take the LTM's to have their synapse weights w_{ij} change according to the following equations:

$$\frac{dw_{ij}}{dt} = \begin{cases} h_j(x_j)\left[-F_{ij}(\boldsymbol{x})w_{ij} + G_{ij}(\boldsymbol{x})f_i(x_i)\right] \\ \quad \text{if top-down } (1 \le i \le k,\ k+1 \le j \le n) \qquad (1a) \\ h_i(x_i)\left[-F_{ij}(\boldsymbol{x})w_{ij} + G_{ij}(\boldsymbol{x})f_j(x_j)\right] \\ \quad \text{if bottom-up } (k+1 \le i \le n,\ 1 \le j \le k). \qquad (1b) \end{cases}$$

Here $\boldsymbol{x}$ is the n-vector of neuron states; $h_l(\cdot)$ and $f_l(\cdot)$ are functions of the state entry x_l and these functions represent the signal strengths that neuron N_l emits when its input is x_l, $h_l(\cdot)$ being for the top neurons and $f_l(\cdot)$ for the bottom ones. A simple choice made in [1, p. 76] and used in the following for $h_l(\cdot)$ and $f_l(\cdot)$ is the identity function, that is, $f_l(x) = x$ and $h_l(x) = x$. Equation (1a) is for top-down LTM where $1 \le i \le k$ and $(k+1) \le j \le n$ while (1b) is for bottom-up LTM where $1 \le j \le k$ and $(k+1) \le i \le n$. For all other ranges of i and j the w_{ij} are absent and, hence, taken to be zero. According to (1), these first-order differential equations can make weight changes only when the top neuron is active (as indicated by its state component being high, making $h_l(\cdot)$ high) and the value of the LTM w_{ij} will grow or decay to the equilibrium value of

$$w_{ij}\Big|_{\text{equi}} = \begin{cases} \dfrac{G_{ij}(\boldsymbol{x})}{F_{ij}(\boldsymbol{x})} f_i(x_i) & \text{for (1a)} \qquad (2a) \\ \dfrac{G_{ij}(\boldsymbol{x})}{F_{ij}(\boldsymbol{x})} f_j(x_j) & \text{for (1b).} \qquad (2b) \end{cases}$$

Because the bottom-up LTM and the top-down LTM have different purposes, as explained in the introductory paragraph, they follow different laws for their weight changes, in which case the choice of F_{ij} differs for $1 \le i \le k$ from that for $k+1 \le i \le n$. We first discuss the bottom-up changes in subsection A and return to top-down ones in subsection B.

A. Bottom-Up LTM

The bottom-up LTM synapse weights, w_{ij}, described by (1b), are used to gate (multiply) the bottom-up signals; that is, the signal flowing into neuron N_i in the F_T field from neuron N_j in the F_B field is the product of w_{ij} and $f_j(x_j)$. The equilibrium value of bottom-up w_{ij}, $G_{ij}(\boldsymbol{x})f_j(x_j)/F_{ij}(\boldsymbol{x})$, was taken in the ART of [1, p. 78] to be in the form of *Weber's law*, that is, using $S_j = f_j(x_j)$,

$$w_{ij}\Big|_{\text{equi}} = \frac{c}{a + \sum_{m=1}^{k} S_m} \qquad (3a)$$

where a and c are positive constants and

$$\sum_{m=1}^{k} S_m = \text{the total amount of the signal comming out of } F_B \text{ neurons.} \qquad (3b)$$

To satisfy the Weber's law equation, Carpenter and Grossberg chose the parameter $F_{ij}(\boldsymbol{x})$ to be [1, p. 75]

$$F_{ij}(\boldsymbol{x}) = f_j(x_j) + L^{-1} \sum_{\substack{m \ne j \\ 1 \le m \le k}} f_m(x_m), \qquad L > 1 \qquad (4)$$

where $1 \le j \le k$, $k+1 \le i \le n$, and $f(\cdot)$ is the function used in the weight change of (1). With this choice of $F_{ij}(\boldsymbol{x})$ and (2) we get, for LTM_{BU},

$$w_{ij}\Big|_{\text{equi}} = \frac{G_{ij}(\boldsymbol{x})}{F_{ij}(\boldsymbol{x})} f_j(x_j) = \frac{G_{ij}(\boldsymbol{x})}{f_j(x_j) + L^{-1} \sum_{\substack{m \ne j \\ 1 \le m \le k}} f_m(x_m)} f_j(x_j)$$

$$= \frac{LG_{ij}(\boldsymbol{x})}{\sum_{1 \le m \le k} f_m(x_m) + (L-1)f_j(x_j)} f_j(x_j). \qquad (5)$$

If x_j is high, that is, if the bottom field neuron N_j has $f_j(x_j) \approx 1$ (where we have normalized "high" of $f(\cdot)$ to 1), we see that (5) gives

$$w_{ij}\Big|_{\text{equi}} \approx \frac{c}{a + (\sum S(x_B))} \qquad \text{(bottom-up)} \qquad (6a)$$

where

$$c = LG_{ij}(\boldsymbol{x}) \tag{6b}$$

$$a = L - 1 \tag{6c}$$

$$\sum S(x_B) = \sum_{m=1}^{k} S_m \tag{6d}$$

$$S_m = f_m(x_m). \tag{6e}$$

Here following [1, p. 74], we choose $G_{ij}(\boldsymbol{x})$ to be constant, that is, independent of $\boldsymbol{x}$, so that the c in (6a) and (6b) will be constant. For x_j low, $f_j(x_j) \approx 0$, we get

$$w_{ij} \approx 0. \tag{6f}$$

The complex choice of $F_{ij}(\boldsymbol{x})$ given by (4) is thus seen to make the equilibrium value of bottom-up w_{ij} in the form of the Weber's law equation. But it also makes the hardware difficult to implement for bottom-up LTM w_{ij} using standard circuits. In the following sections in order to rather easily implement a close approximation to this type of w_{ij}, we introduce an "axon-synapse-tree" structure, which is based on the shape of a biological neuron's axon and synapse.

1) Serially Connected Voltage-Controlled Resistors: For an artificial neuron to generate a function of the form needed at (6a), that is, $c/(a + b(x))$, it is not practical to sum up a and b and then take the whole sum to divide c. In a real neuron, every function is generated so naturally that the complex computations are inherent. In this view we look for a natural way to implement the function of (6).

As the clue to our implementation, we note that Hartline's neural chemical pool of type 1, which acts as a bound transmitter at a synapse [4, p. 658], follows the law $c/(a + b(x))$. This can be generated by a series connection of voltage-controlled resistors where the conductances of these resistors are proportional to the control voltages V_a and V_b as shown in Fig. 3. In simulating a chemical pool the conductances of these two resistors, R1 and R2, are linearly modulated by the concentrations of chemical materials, denoted V_a and V_b in Fig. 3. The conductances are K_aV_a and K_bV_b, respectively.

The current I, which flows through R1 and R2, is

$$I = \frac{V_d}{\frac{1}{K_aV_a} + \frac{1}{K_bV_b}} = \frac{V_d}{\frac{K_bV_b + K_aV_a}{K_aK_bV_aV_b}} = \frac{K_aK_bV_dV_aV_b}{K_aV_a + K_bV_b}. \tag{7}$$

If the voltage V_s is taken at the junction of R1 and R2,

$$V_s = I \cdot \frac{1}{K_bV_b} = \frac{K_aK_bV_dV_aV_b}{K_aV_a + K_bV_b} \cdot \frac{1}{K_bV_b}$$

$$= \frac{V_dV_a}{V_a + \frac{K_b}{K_a} \cdot V_b} = \frac{c}{a + b} \tag{8a}$$

where

$$a = V_a \tag{8b}$$

$$b = \frac{K_b}{K_a} \cdot V_b \tag{8c}$$

and

$$c = V_dV_a. \tag{8d}$$

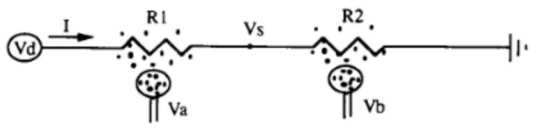


Fig. 3. A series connection of voltage-controlled resistors to generate the function $c/(a + b)$.

This series connection of resistors generates the Weber's law function as long as the b in (8a) is the function ΣS in (6a). Equation (8a) is under the assumption that the conductances of R1 and R2 are linearly proportional to the voltages V_a and V_b. This linearity is difficult to achieve by using present-day MOS devices as resistors. But a single MOS resistor with its conductance nonlinearly modulated by its gate voltage can be used to produce a function which is very similar to the function in Weber's law, and this can be used to obtain a satisfactory implementation of ART bottom-up LTM's. In the next section we will use an Axon-synapse-tree (AST) to approximate Weber's law.

2) Axon-Synapse-Tree Structure: By the discussion of subsection III-A-1, we can now develop an AST MOS structure to implement the bottom-up LTM.

A biological neuron can transmit signals from its cell body through its axon to synapses, and the conductances of the synapses are subject to change owing to concentrations of chemical materials. Considering a synapse, its conductance can be modulated by some pool level (concentration of chemical materials) [4], [5]. The higher the concentration of chemical material, the higher the conductance of the synapse. We can then build the "AST" circuit of Fig. 4, which is based on a neuron's shape while all the synapses and axon resistors are realized by MOS transistors which operate in their ohmic region and whose conductances are controlled by their gate voltages S_i, where S_i is the signal coming from STM_B as per (6e). Because the ART1 STM transfers rapidly between "high" and "low" states, we can say that the strength of the signal S_i controlling the synapse conductance is either S_{high} or S_{low}. In order to turn on the transistors in the AST synapses, S_{high} has to be greater than V_t while S_{low} is less than V_t.

By definition, we let the nonlinear conductance of an MOS transistor in its ohmic region be its drain current divided by its drain-to-source voltage. That is,

$$G = \frac{I_d}{V_{ds}} = \frac{\beta\left[(V_{gs} - V_t)V_{ds} - (V_{ds}^2/2)\right]}{V_{ds}}$$

$$= \beta\left[(V_{gs} - V_t) - \frac{V_{ds}}{2}\right] \tag{9a}$$

where

$$V_{gs} > V_t > 0 \qquad V_{gs} - V_{ds} > V_t \qquad \beta = \frac{\mu C_{\text{ox}} W}{L}. \tag{9b}$$

Because all the AST synaptic branches are connected in parallel, assuming all the MOS transistors in the AST synapses are identical, all these synapse resistors can be lumped into a single equivalent resistor whose conductance is $p\beta(S_{\text{high}} - V_t - V_s/2)$, where p denotes the number of input signals that are high (Fig. 4). By this conversion the AST structure is now the same as that of the circuit in Fig. 3 except that the conductances of R1 and R2 are now $\beta(V_a - V_s - V_t - (C - V_s)/2)$ and

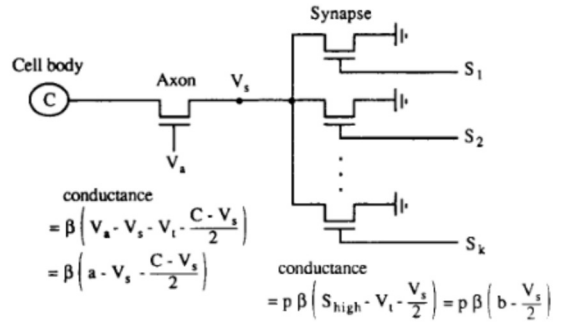


Fig. 4. The electronic circuit to model the neuron with axon and synapses. S_j, $1 \le j \le k$, are the voltages of control signals. Here we design the system such that all the transistors are in their ohmic region. V_a is a constant.

$p\beta(S_{high} - V_t - V_s/2)$, respectively, where C is the constant excitation. Because V_a and S_{high} are constant voltage levels, we can, without loss of generosity, assume the conductances of the resistors to be $\beta(a - V_s - (C - V_s)/2)$ and $p\beta(b - V_s/2)$, respectively, where $a = V_a - V_t$ and $b = S_{high} - V_t$ are constants.

Applying (8a) to the AST structure, we can get the voltage V_s:

$$V_s = \frac{C[a - V_s - (C - V_s)/2]}{[a - V_s - (C - V_s)/2] + p\dfrac{\beta}{\beta}(b - V_s/2)} = \frac{C[a - V_s - (C - V_s)/2]}{(a + pb) - [C + (p + 1)V_s]/2} \tag{10}$$

By solving (10) we get the output voltage, V_s, as a function of p (the number of synapse transistors that are turned on):

$$V_s(p) = \frac{(a + pb) - \sqrt{(a + pb)^2 - 2(p + 1)\left(aC - \dfrac{C^2}{2}\right)}}{p + 1} \tag{11}$$

Comparing the function $V_s(p)$ in (11) generated by the AST with the function $W(p) = c/(a + pb)$ (a, b, and c being constants and p a variable) from Weber's law, we find that they both have the same properties; that is, $V_s(p)$ and $W(p)$ are decreasing functions of p while both $p \cdot V_s(p)$ and $p \cdot W(p)$ are increasing functions (Fig. 5). Following [1, pp. 78–79] and replacing their Weber's law by our function $V_s(p)$ resulting from the AST structure, we can show the validity of direct access to subset and superset storage in the ART with the AST structure in bottom-up LTM without a hardware implementation of Weber's law function.

A PSPICE simulation of an AST is shown in Figs. 6 and 7. Fig. 6 shows an AST with five synapse resistors where the synapse resistors of the AST are realized by the MOS resistors described by (9). The voltage of the output node, marked as node 1 in Fig. 6, is shown in Fig. 7. The transistor lengths and widths used are set at 5 μm each in the simulation, although the size of the transistors is not critical in the AST as long as they are essentially the same. The simulations here and following are based on the SPICE level2 parameters of MOSIS SCPE run on

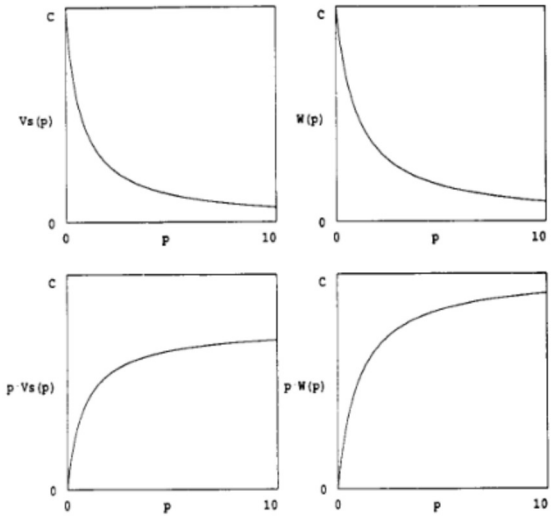


Fig. 5. Comparisons between the functions $V_s(p)$ and $W(p)$. $V_s(p)$ and $W(p)$ are both decreasing functions of p while both $pV_s(p)$ and $pW(p)$ are increasing functions of p.

2/8/90 (including for n-channel VT0 = 1.068, Kp = 5.018E-5 and for p-channel VT0 = −0.73, Kp = 2.007E-5).

3) VLSI Implementation of Bottom-Up LTM: The implementation of bottom-up LTM is straightforward with the AST structure on hand. From the properties of ART LTM, w_{ij} can change its value when only x_i, $k + 1 \le i \le n$, is activated. Also w_{ij} will decay to zereo when x_j, $1 \le j \le k$, is low and grows to $V_s(p)$ when x_j is high [1], [2]. In other words, for i indexing a top neuron and j a bottom neuron.

$$w_{ij} \to V_s \quad \text{when } x_i \text{ is high and } x_j \text{ is high} \tag{12}$$

$$w_{ij} \to 0 \quad \text{when } x_i \text{ is high and } x_j \text{ is low} \tag{13}$$

while w_{ij} does not change when x_i is low.

These gating properties suggest that bottom-up w_{ij} can be implemented by transmission gates, as in Fig. 8, where w_{ij} is measured as the voltages on the capacitors. In this figure, M1, M2, M3, and M4 are n-type MOS transmission gates. The signal of $V_s(p)$ will pass to w_{ij} when x_i and x_j are both high, and w_{ij} will decay to 0 if x_i is high and x_j is low. Because n-type transmission gates can transmit signals perfectly only when the transmitted signal is less than the gate voltage by V_t [6, p. 7], the following inequality has to be met to ensure proper transmission:

$$x_i(\text{high}) - V_t > V_s(p) \tag{14}$$

$$x_j(\text{high}) - V_t > V_s(p). \tag{15}$$

Because we choose $x_i(\text{high}) = x_j(\text{high}) = 5$ V and there is at least one x_j high when the signal is transmitted, the voltage $V_s(p)$ generated by the AST is less than or equal to $V_s(1)$. We have

$$V_s(p) \le V_s(1) \approx 3 \text{ V} \quad \text{(Fig. 7)}$$

$$\Rightarrow V_{i(\text{or } j)}(\text{high}) - V_t > V_s(p). \tag{16}$$

This shows that all LTM w_{ij} end up being V_s or 0.

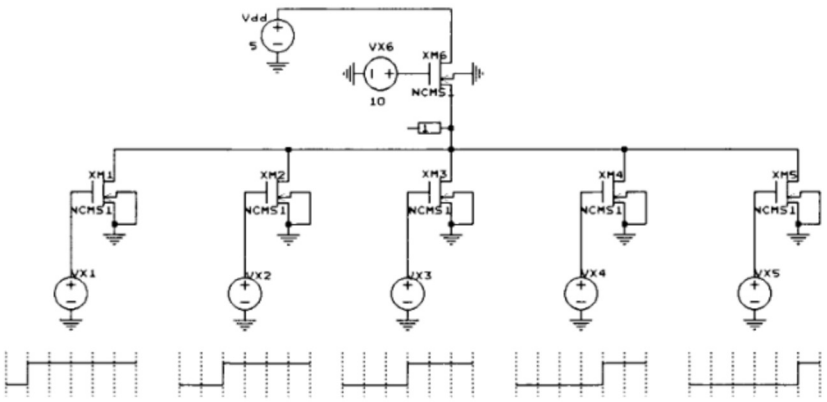


Fig. 6. Circuit for axon-synapse tree to generate the function $V_s(p)$ to mimic the Weber's law function. The "high" for the F_B neuron states x_i is set at 5 V; the "low" is set at 0 V.

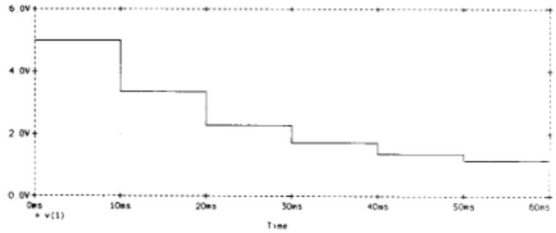


Fig. 7. Simulation result of axon-synapse tree. $V_s(p) = V(1)$ is a decreasing function of p, where p, $p = 0, \cdots, 5$, is the number of "high" input signals for Fig. 6.

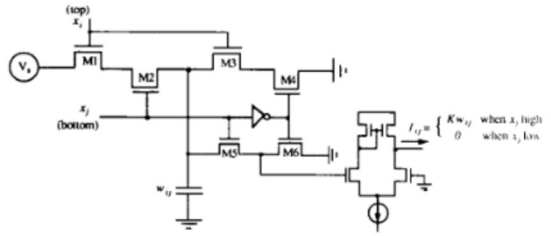


Fig. 8. Transmission gate implementation of bottom-up synapse strength. Transistors M1, M2, M3, and M4 are used to determine whether w_{ij} should increase to V_s or decay to 0 while M5 and M6 are used to control the transmission of w_{ij}, which in turn control the input voltage of the differential pair. In this structure the output current $I_{ij} = 0$ when x_j is low and $I_{ij} = Kw_{ij}$ when x_j is high, K being a constant representing the gain of the differential pair.

The bottom-up signal, which is represented by a current I_{ij} in Fig. 8, is implemented by a differential pair with n-type MOS transmission gates to control one of the input voltages of the differential pair to ensure that I_{ij} is proportional to w_{ij} when x_j is high and equal to 0 when x_j is low.

Fig. 9 shows a circuit for a 4 by 1 bottom-up array of synapse weights that functions between one F_T neuron N_i, say N_5, and four F_B neurons, N_1, N_2, N_3, and N_4. The voltages $V(51)$, $V(52)$, $V(53)$, and $V(54)$ on the capacitors C_{51}, C_{52}, C_{53}, and C_{54} represent the synapse weights w_{51}, w_{52}, w_{53}, and w_{54}, respectively. Simulation results are shown in Fig. 10, where these voltages, representing the LTM's, are specified by (12) and (13). The four bottom STM's and one top STM are represented by the input voltages $V(1)$, $V(2)$, $V(3)$, $V(4)$, and $V(5)$ shown in Fig. 9. $V(10)$ is the voltage of $V_s(p)$ generated by AST. By examining the output curves in Fig. 10, the top STM $V(10)$ goes high at 20 μs, so that $w_{51} = V(51)$ decays to 0 because $V(1)$ is low. The weights w_{52}, w_{53}, and w_{54} will grow to $V_s(3)$ because $V(2)$, $V(3)$, and $v(4)$ are high and there are three bottom neurons that are high.

B. Top-Down LTM

As shown in (2), the LTM equilibrium weights satisfy the same expression for both bottom-up and top-down. But in Carpenter and Grossberg's model [1, p. 75], the top-down LTM does not follow Weber's law. The law for top-down LTM is that the w_{ij}, $1 \le i \le k$ and $k + 1 \le j \le n$, grow to their high values when x_j and x_i are high while the w_{ij} decay to zero when x_j is high and x_i is low. When x_j is low, no change occurs. This suggests that F_{ij} and G_{ij} for (1a) and (2a) are chosen to be constants. Thus, the implementation of the bottom-up LTM given in Fig. 8 is also the circuit to implement top-down LTM, except that the voltage source V_s in Fig. 10 should now be replaced by a constant voltage source which represents the "high" value for the top-down synapse weights w_{ij}.

IV. Short-Term Memory

STM is the state x_i of the neurons themselves in the combined F_B and F_T fields. In ART every neuron receives a bunch of signals, some of which are excitatory inputs and some of which are inhibitory inputs. To avoid the case where a neuron state x_i grows to infinity because it receives mostly excitatory signals or the case where x_i decays to negative infinity when inhibitory inputs are much larger than excitatory inputs, a shunting STM law is needed for the ART STM's. Such a law is given in [3, p. 23] and is, for all STM neurons N_i, $1 \le i \le n$,

$$\frac{d}{dt} x_i = -A_i x_i + (B_i - C_i x_i) J_i^+ - (E_i + F_i x_i) J_i^- \quad (17)$$

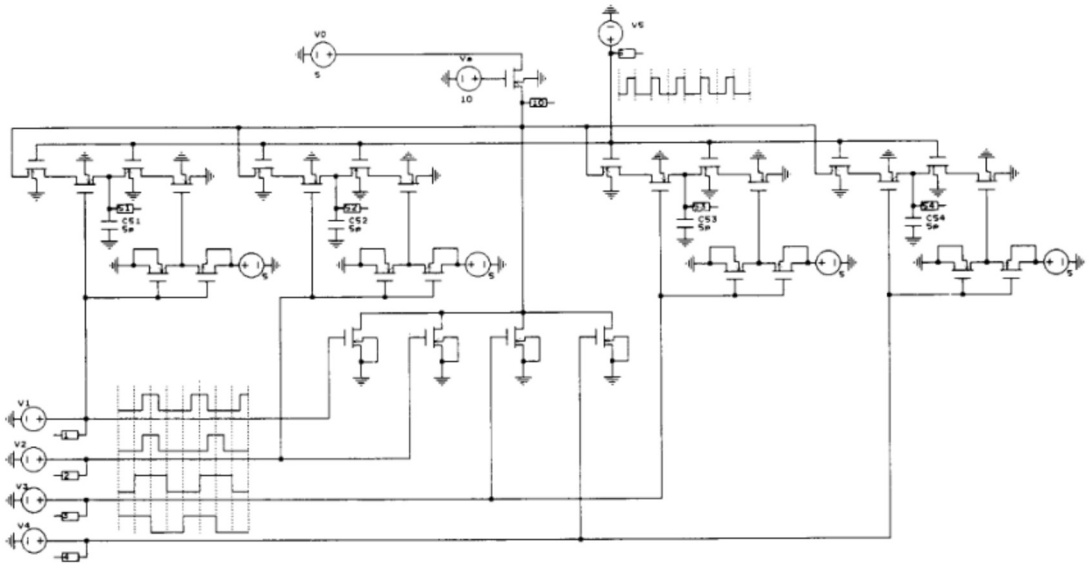


Fig. 9. Circuit of a 4 by 1 bottom-up LTM for four F_B neurons and one F_T neuron. The voltages on capacitors C_{ij}, $1 \le j \le 4$, represent the bottom-up LTM from N_j in F_B to N_i in F_T. The time scale for input signals $V(1)$–$V(5)$ is 0 up to 80 μs.

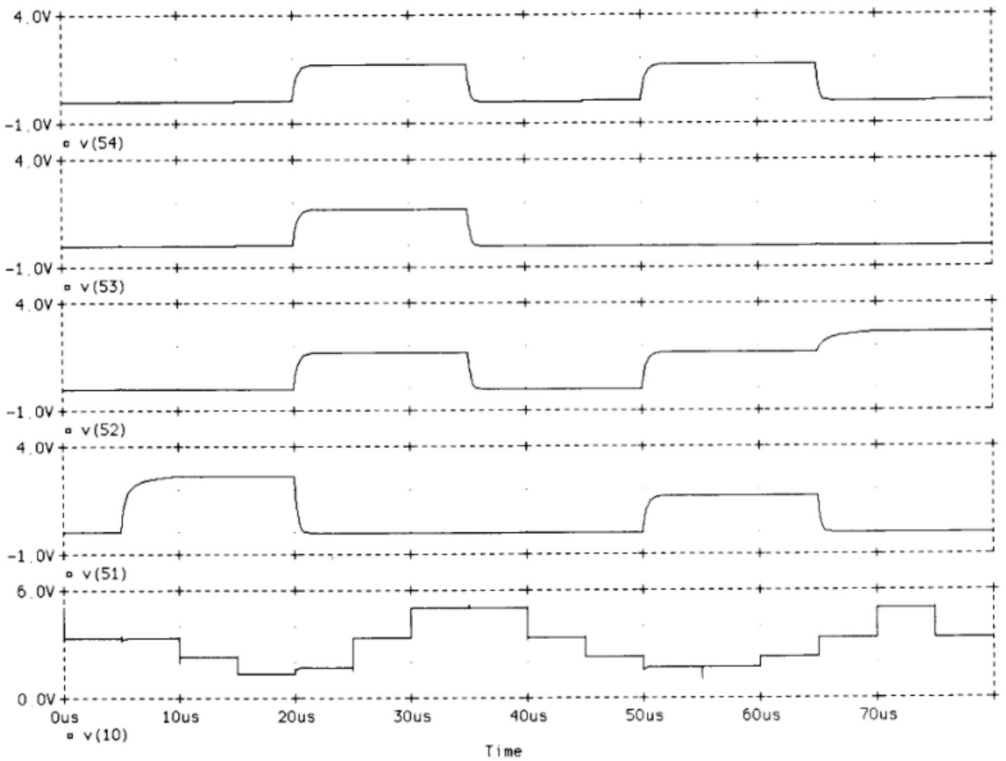
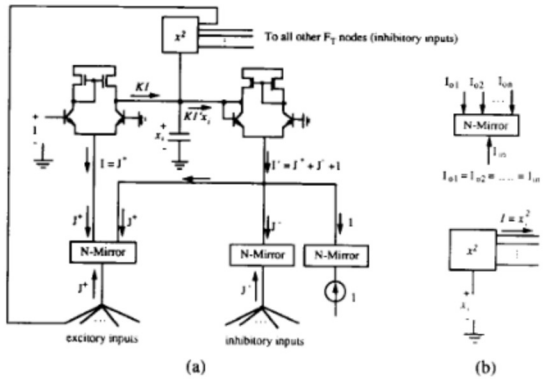


Fig. 10. Simulation results of Fig. 9. $V(51)$, $V(52)$, $V(53)$, and $V(54)$ are the voltages representing the bottom-up LTM with the states $V(1)$–$V(5)$ of bottom and top neurons shown in Fig. 9.



(a) (b)

Fig. 11. (a) Structure of shunting STM's in the F_T field assuming the range of x_i for this STM is [0, 1]. For STM's in the F_B field, remove the "square" function on the top. (b) The symbolism for the current mirror and the analog squaring device.

where J_i^+ is the sum of excitatory inputs and J_i^- is the sum of inhibitory inputs for neuron N_i. A_i, B_i, C_i, E_i, and F_i are constants. In this equation the STM state x_i is bound to the interval $[-E_i/F_i, B_i/C_i]$ because when $x_i > (B_i/C_i)$, the excitatory inputs become inhibitory and when $x_i < (-E_i/F_i)$, all the inhibitory inputs become excitatory. A_i will be chosen large so that the STM can change its values much faster than the LTM does.

In a binary input neural network, two ranges of STM's are of interest, these being [0, 1] and [−1, 1]. If the range for the STM x_i is [−1, 1], and $A_i = A$ for all i, (17) can be rewritten as

$$\frac{1}{A}\frac{d}{dt}x_i = -x_i + (1 - x_i)J_i^+ - (1 + x_i)J_i^- \quad (18a)$$

$$= (J_i^+ - J_i^-) - (1 + J_i^+ + J_i^-)x_i \quad (18b)$$

$$= C - C'x_i \quad (18c)$$

where

$$C = J_i^+ - J_i^- \quad \text{and} \quad C' = 1 + J_i^+ + J_i^-. \quad (19)$$

On the other hand if the range for STM is [0, 1], (18c) again results but (19) is replaced by

$$C = J_i^+ \quad \text{and} \quad C' = 1 + J_i^+ + J_i^- \quad (20)$$

Equation (18c) is actually that describing a type-4 chemical pool, which functions as a second messenger in Hartline's neural networks [4, p. 658]. The chemical pool described by (18c) has been realized by a VLSI circuit [5]. In that realization, a pool itself can be viewed as a capacitor and the voltage on the capacitor is the pool concentration (concentration of chemical materials) which we now use to represent the STM of the ART. If the capacitance (volume of pool) is $1/A$, the term $(1/A)(d/dt)\,x_i$ represents the rate of change of chemical materials in the pool. This rate can be realized in a circuit as the sum of currents flowing into and out of the capacitor. The STM then can be implemented by the circuit of Fig. 11 [5], where the C and C' in (20) are represented by the currents I and I'.

Fig. 11 is a schematic circuit for a shunting STM with its potential range in [0, 1]. If the range STM is desired to be [−1, 1], we can add an additional differential pair to draw an additional current KJ^- out of the capacitor. The "square" circuit on top of the circuit is for F_T neurons to perform "competition" and should be omitted when representing the F_B neurons. For F_T neurons the inhibitory inputs come from the reset subsystem (Fig. 1) and all other F_T neurons. For F_B neurons the inhibitory inputs come from the gain control subsystem (Fig. 1) [1], [2]. Because only "memory" structures are discussed here, we do not go any further into these inhibitory functions.

V. Discussion

The ART memories, including LTM's and STM's, are implemented by VLSI circuits in which these memories are viewed as the concentrations of chemical materials that serve as bound transmitters and second messengers in biological neurons. An axon-synapse-tree structure is introduced to achieve the properties of Weber's law in the bottom-up LTM. The $f_j(x_j)$ which assist in adjusting LTM from STM are realized in our circuit by the identity function; that is, $f(x) = x$. Because of the constraints that the MOS resistors of Fig. 4 turn on only when the control voltage (gate voltage), $f(x)$, is higher than $V_s + V_t$, only memories for binary input ART are implemented with the circuits of this paper. Analog input ART memories are still under study.

A 4 by 4 bottom-up LTM has been layed out using Magic, with the capacitors being made 5 pF by double-poly technology. The charging time constant can be changed to fit the desired operation of ART by altering the size of these capacitors or by changing the lengths of the gating transistors. Also, we note that the AST structure in bottom-up LTM is not very sensitive to a slight mismatch of transistors since we still obtain the same desired properties that $V_s(p)$ and $p \cdot V_s(p)$ decrease and increase, respectively, with p. For VLSI we desire devices as small as possible, in which case short-channel transistors are convenient. It appears again that the overall behavior is about the same with short-channel devices, but that remains to be verified.

It should be noted that LTM is stored as voltages in the capacitors of Fig. 8. Because of leakage, this LTM is subject to degradation. Consequently, means of leak-free capacitive storage should be sought, for example through new classes of floating-gate devices (with low write voltages) or through added refresh circuitry. This is one of the many challenging problems still to be faced in making ART hardware realization practical.

Acknowledgment

The authors wish to express their appreciation to the reviewers, whose comments were very helpful in improving the final paper.

References

[1] G. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Comput. Vision, Graphics, and Image Process.*, Vol. 37, pp. 54–115, 1987.

[2] G. Carpenter and S. Grossberg, "ART2: Self-organization of stable category recognition codes for analog input patterns," *Appl. Opt.*, vol. 26, no. 23, pp. 4919–4930, December 1987.

[3] S. Grossberg, "Nonlinear neural networks: Principles, mechanisms, and architectures," *Neural Networks*, vol. 1, pp. 17–61, 1988.

[4] D. Hartline, "Simulation of restricted neural networks with reprogrammable neurons," *IEEE Trans. Circuits and Syst.*, vol. 36, pp. 653–660, May 1989.
[5] S. W. Tsay, N. El-Leithy, and R. W. Newcomb, "CMOS realization of a class of Hartline neural pools," in *Proc. IEEE Int. Symp. Circuits Syst.* (New Orleans), May, 1990, pp. 2417–2420.
[6] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*. Redwood City, CA: Addison-Wesley, 1985.
[7] P. Gray and R. Meyer, *Analysis and Design of Analog Integrated Circuit*, 2nd ed. New York: Wiley, 1984.

Suan Wei Tsay received the B.S. degree from National Taiwan University, Taipei, Taiwan, in 1984, and the M.S. degree from the University of Maryland, College Park, in 1988, both in electrical engineering. He is currently working toward the Ph.D. degree at the University of Maryland. His research interests encompass neural networks, analog/digital VLSI circuits, and parallel processing.

Robert W. Newcomb (S'52–M'56–F'72) was born in Glendale, CA, in June 1933. He received the BSEE from Purdue in 1955, the MS from Stanford in 1957, and the Ph.D. from the University of California, Berkeley, in 1960, all in electrical engineering.

He was a Research Intern at the Stanford Research Institute from 1955 to 1957 and was on the faculties of Berkeley (1957–1960), and Stanford (1960–1970), before joining the University of Maryland in 1970 to direct the graduate program in electrical engineering.

Dr. Newcomb is a registered Professional Engineer in the state of California. He has had Fulbright Fellowships to Australia and Malaysia and leaves to Belgium and Spain. He has chaired a number of international meetings in the systems theory area, is a founder of the Mathematical Theory of Networks and Systems International Symposium, a founding member of the IEEE Council on Robotics and Automation, and an AdCom member of the IEEE Society for Social Implications of Technology and the IEEE Neural Networks Council. He is active in the IEEE Circuits and Systems Society, where he has been Chairman of the IEEE CAS Technical Committee on Neural Systems and Applications. He has been an external examiner and program evaluator for a number of universities around the world. Recently in Spain he assisted in setting up a robotics laboratory while organizing the research faculty "Grupo de Trabajo en Sistemas PARCOR," directing its activities primarily in the biomedical signal processing area within the computer faculty of the Universidad Politécnica de Madrid. His present research concentrates upon nonlinear semistate theory and its applications to such areas as neural-type microelectronics, biomedical signal processing including noninvasive Kemp echo determination of ear parameters, and robotics. He awards the Academy of American Poets Prize at the University of Maryland as well as the Z. Aziz Fellowship for students in biomedical engineering.