INT. J. ELECTRONICS, 1994, VOL. 76, NO. 2, 271–291

Analogue circuit design and implementation of an adaptive resonance theory (ART) neural network architecture

CHING S. HO†, JUIN J. LIOU†, MICHAEL GEORGIOPOULOS†, GREGORY L. HEILEMAN‡ and CHROSTOS CHRISTODOULOU†

An analogue circuit implementation is presented for an adaptive resonance theory neural network architecture, called the augmented ART-1 neural network (AART1-NN). The AART1-NN is a modification of the popular ART1-NN, developed by Carpenter and Grossberg, and it exhibits the same behaviour as the ART1-NN. The AART1-NN is a real-time model, and has the ability to classify an arbitrary set of binary input patterns into different clusters. The design of the AART1-NN circuit is based on a set of coupled nonlinear differential equations that constitute the AART1-NN model. The circuit is implemented by utilizing analogue electronic components such as operational amplifiers, transistors, capacitors, and resistors. The implemented circuit is verified using the PSpice circuit simulator, running on Sun workstations. Results obtained from the PSpice circuit simulation compare favourably with simulation results produced by solving the differential equations numerically. The prototype system developed here can be used as a building block for larger AART1-NN architectures, as well as for other types of ART architectures that involve the AART1-NN model.

1. Introduction

Information processing and management in modern military and commercial systems is growing more complex and is requiring higher performance. Traditional computational approaches have not met the challenge posed by these expanding requirements. Neural networks offer new computational solutions to these requirements that are potentially robust to noise and incomplete information, adaptive to a changing environment, intrinsically massively parallel and fault tolerant. Neural networks are currently realized in a number of ways. Most of the neural networks in the open literature have been implemented via computer simulation of their corresponding mathematical models. For real-life applications, however, neural networks need to be implemented as analogue, digital, or hybrid (analogue digital) hardware.

In this paper we focus our attention on the hardware implementation of an adaptive resonance theory neural network named AART1-NN (augmented adaptive resonance theory-1 neural network). The AART1-NN was developed by Heileman *et al.* (1992), and is a modification of the popular ART1-NN introduced by Carpenter and Grossberg (1987). The major difference between the AART1-NN and the ART1-NN is that the AART1-NN is completely described by a set of differential equations, while the ART1-NN incorporates algorithmic components in its description. As Carpenter (1989) points out, differential equations constitute the language of real-time models. Hence, from this perspective the AART1-NN is a real-time

Received 2 September 1993; accepted 7 September 1993.

†Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL 32816, U.S.A.

‡Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87185, U.S.A.

model. The meaning of *real-time* is twofold: in the neural network literature, it describes systems that require no external control of system dynamics. In the engineering literature, it is used to describe a system that is able to process inputs as fast as they arrive. There is no doubt that it is advantageous for a system such as the AART-NN to have the aforementioned characteristics. It is also worth pointing out that the AART1-NN exhibits a behaviour identical with the behaviour of the ART1-NN (for a proof see Carpenter and Grossberg 1987). Since both models exhibit identical behaviour, a hardware implementation of the AART1-NN will provide us with a hardware module that behaves like the ART1-NN.

Let us now elaborate on the significance of designing hardware that guarantees an operation identical with that of the ART1-NN. The novel property of the ART1-NN is the controlled discovery of clusters in response to binary input patterns presented to it. In addition, the ART1-NN can accommodate new clusters without affecting the storage or recall capabilities for clusters already learned. To date, the ART1-NN has been used successfully in a variety of applications. In particular, the Boeing Corporation (Caudell 1991) has utilized the ART1-NN, or modifications of it, for automatic target recognition, sensor and data fusion, and aircraft design retrieval applications; in all of these cases, real-world data have been used. Furthermore, the ART1-NN has been utilized in the design of larger ART architectures, such as ARTMAP (Carpenter *et al.* 1991) and Fuzzy ARTMAP (Carpenter *et al.* 1992), to mention only two. These architectures exhibit capabilities beyond those achieved by the ART1-NN alone. For example, supervised learning is possible with ARTMAP and Fuzzy ARTMAP, and consequently these architectures can be used as pattern recognition machines.

In our paper an analogue circuit is designed and implemented based on the set of differential equations tht describe the AART1-NN model. One of the primary reasons that implementation of neural networks on analogue hardware is a worthwhile pursuit stems from the fact that a large proportion of neutral computation in the human brain is done in an analogue rather than a digital manner. The building blocks utilized to implement the AART1-NN model in this work are weighted summers, comparators, multipliers, and integrators. All of these blocks are implemented using operational amplifiers. Operational amplifiers are relatively inexpensive and they have been used in a variety of circuit designs, some of them involving neural networks (Fryre *et al.* 1991, Read and Geiger 1989). There have been other efforts in the past to implement an ART1-NN in hardware. For example, Nahet *et al.* (1989) developed an analogue implementation of shunting neural networks. The equations that describe short term memory (STM) activities in the AART1-NN (for more details see § 2) are shunting equations. However, no attempt was made by Nahet *et al.* (1989) fully to implement a compete adaptive resonance theory neural network architecture, a task that is accomplished in this paper. A VLSI implementation of the ART1-NN has been conducted by Tsay and Newcomb (1991). The particular implementation encompasses the (STM) equations, and the long term memory (LTM) equations. But Tsay and Newcomb (1991) implemented the LTM equations at their equilibrium point, and omitted the implementation of the reset subsystem in the ART1-NN. No such restrictions are imposed in our implementation. As we demonstrate in § 3, the complete adaptive resonance theory neural network is implemented, which yields a circuit that can operate without external supervision The only information that the circuit requires to function

properly is the collection of the patterns that are applied at the neural network input.

The organization of our paper is as follows: in the second section, the AART1-NN model is described as a set of nonlinear differential equations. In the third section, we proceed with the hardware design of a prototype AART1-NN. In §4, we describe qualitatively how the circuit of §3 is supposed to operate. This allows us to test the PSpice simulation results, presented in §5 both qualitatively and quantitatively. The qualitative test is based on the description of §4, while the quantitative test is based on simulation results obtained by numerically solving the differential equations that define the AART1-NN model. The PSpice simulation results turn out to fare well in both tests. Finally, in §6, we give an overview of the paper and provide some concluding remarks.

2. The AART1-NN model

The major components of the AART1-NN model are depicted in Fig. 1. These components can be grouped into two subsystems—the *attentional subsystem* and the *orienting subsystem*. The attentional subsystem consists of two fields of nodes, designated as F_1 and F_2 fields. The nodes in the F_1 and F_2 fields are used to encode patterns of STM activity. Each node in the F_1 field is connected via bottom-up weighted connections, called 'bottom-up LTM traces', to all the nodes in the first layer of the F_2 field. Furthermore, every node in the first layer of the F_2 field is connected via top-down weighted connections, designated as 'top-down LTM traces', to all the nodes in the F_1 field. The bottom-up and top-down LTM connections are adaptive and are used to store the knowledge acquired by the neural network as its training progresses. Finally, the nodes in the first layer of the F_2 field are completely connected via non-adaptive connections. In Fig. 1, the bottom-up

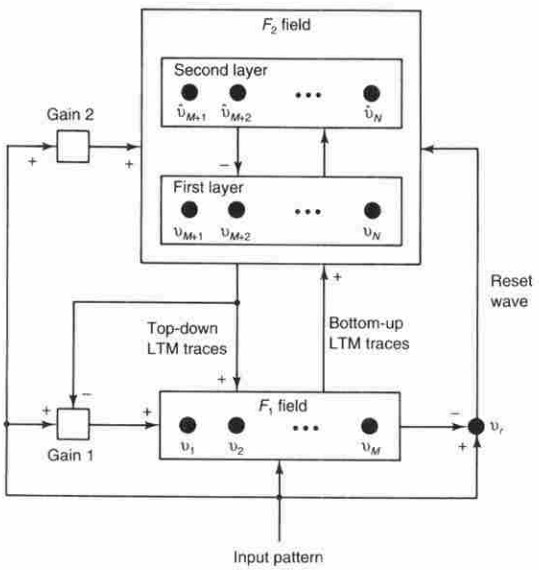


Figure 1. The architecture of the augmented ART1 neural network (AART1-NN) model.

LTM traces are depicted by an arrow from the F_1 field to the F_2 field, while the top-down LTM traces are depicted by an arrow from the F_2 field to the F_1 field. The non-adaptive connections among the nodes in the first layer of the F_2 field are omitted for reasons of figure clarity. The F_1 field is often referred to as the 'input field' because the input patterns are presented to it. The first layer of the F_2 field is often referred to as the 'category representation layer', because it is the layer that indicates the category to which the input pattern belongs. The orienting subsystem in the AART1-NN architecture consists of a single node designated as v_r (see Fig. 1). The primary purpose of the orienting sybsystem is to generate a reset wave to the F_2 field, whenever the category representation in the first layer of the F_2 field is not a good match for the input pattern. The objective of the second layer of nodes in the F_2 field is to deactivate the erroneous category representation in the first layer of the F_2 field, whenever such an erroneous representation occurs, and to keep this erroneous category deactivated for as long as the same input pattern persists at the F_1 field. The AART1-NN architecture is completed by a couple of gain control mechanisms, designated as Gain 1 and Gain 2. The functionality of Gain 1 is to guarantee the satisfaction of the '2/3 rule' (for more details see Carpenter and Grossberg 1987). Gain 2, in collaboration with the second layer of the F_2 field nodes, maintains the deactivation of erroneous category representations in the first layer of the F_2 field for as long as the input pattern persists across the F_1 field; furthermore, Gain 2 drives the STM activity of the network to the resting value of zero in the absence of input patterns at the F_1 field.

We denote nodes in the F_1 field by v_i, nodes in the first layer of the F_2 field by v_j and nodes in the second layer of the F_2 field by $\hat{v}_j$. As we mentioned before, the reset node is designated as v_r. The index of nodes in the F_1 field ranges from 1 to M, while that of nodes in the first and second layer of the F_2 field ranges from $M+1$ to N. We denote the activities of the nodes v_i, v_j and $\hat{v}_j$ by x_i, x_j and $\hat{x}_j$, respectively. The activity of a node v_i in the F_1 field is described by the following differential equation:

$$\varepsilon_1 \frac{\mathrm{d}}{\mathrm{d}t} x_i = -x_i + (1 - A_1 x_i) J_i^+ - (B_1 + C_1 x_i) J_i^- \tag{1}$$

where J_i^+ and J_i^- represent the total excitatory and inhibitory input to the node v_i, respectively. In particular, J_i^+ and J_i^- are given by the following equations:

$$J_i^+ = I_i + D_1 \sum_{j=M+1}^{N} f_2(x_j) z_{ji} \tag{2}$$

$$J_i^- = \sum_{j=M+1}^{N} f_2(x_j) \tag{3}$$

In (2) and (3), A_1, B_1, C_1 and D_1 are constants, I_i is the component of the binary input pattern I that is received by node v_i, $f_2(x_j)$ is the output activity generated by node v_j with activity x_j, and z_{ji} is the value of the top-down LTM trace between node v_j in the F_2 field and node v_i in the F_1 field. In (2) and (3) and throughout this paper we assume that the output activity generated by a node v_j with activity x_j is the threshold function

$$f_2(x_j)=\begin{cases}1 & \text{if } x_j>\delta_2\\ 0, & \text{otherwise}\end{cases} \tag{4}$$

where δ_2 is the threshold of every node v_j in the first layer of the F_2 field. The meaning of the threshold is that if the node activity exceeds the threshold value the output activity of the node is positive (i.e. the node is active), otherwise the output activity of the node is zero (i.e. the node is inactive). The activity of a node v_j in the first layer of the F_2 field satisfies the following differential equation:

$$\varepsilon_2\frac{\mathrm{d}}{\mathrm{d}t}x_j=-x_j+(1-A_2x_j)J_j^+-(B_2+C_2x_j)J_j^- \tag{5}$$

where J_j^+ and J_j^- represent the total excitatory and inhibitory input to the node v_j, respectively. In particular, J_j^+ and J_j^- are given by the following equations:

$$J_j^+=f_2(x_j)g(I)+D_2\sum_{i=1}^{M}f_1(x_i)z_{ij} \tag{6}$$

$$J_j^-=\sum_{k\neq 1}f_2(x_k)+\hat{f}_2(\hat{x}_j) \tag{7}$$

In (6) and (7), A_2, B_2, C_2 and D_2 are constants, $f_1(x_i)$ is the output activity generated by node v_i with activity x_i, $f_2(\hat{x}_j)$ is the output activity produced by node $\hat{v}_j$ with activity $\hat{x}_j$, and z_{ij} is the value of the bottom-up LTM trace corresponding to the connection between node v_i in the F_1 field and node v_j in the F_2 field. The output activities $f_1(x_i)$ and $\hat{f}_2(\hat{x}_j)$, as well as the function $g(I)$, appearing in (6), are provided by the following equations:

$$f_1(x_i)=\begin{cases}1 & \text{if } x_i>\delta_1\\ 0, & \text{otherwise}\end{cases} \tag{8}$$

$$\hat{f}_2(\hat{x}_j)=\begin{cases}1 & \text{if } \hat{x}_j>\hat{\delta}_2\\ 0, & \text{otherwise}\end{cases} \tag{9}$$

$$g(I)=\begin{cases}1 & \text{if } \sum_{i=1}^{M}I_i\neq 0\\ 0, & \text{otherwise}\end{cases} \tag{10}$$

In (8) and (9), δ_1 and $\hat{\delta}_2$ represent the thresholds of nodes in the F_1 field and the second layer of the F_2 field, respectively. The meaning of these thresholds is the same as the meaning of the threshold δ_2 that we discussed above. The activity $\hat{x}_j$ of a node $\hat{v}_j$ is provided by the following differential equation:

$$\varepsilon_2\frac{\mathrm{d}}{\mathrm{d}t}\hat{x}_j=-[1-g(I)]\hat{x}_j+g(I)f_r(x_r)f_2(x_j] \tag{11}$$

In the above equation, $f_r(x_r)$ stands for the output activity of the reset node v_r with activity x_r. The activity x_r of the reset node is given by the following differential equation:

$$\varepsilon_r \frac{d}{dt} x_r = -A_r x_r + U\left[P \sum_{i=1}^{M} I_i - Q \sum_{i=1}^{M} f_1(x_i)\right] \tag{12}$$

where A_r, P and Q are constants and U stands for the unit step function (i.e. $U(x)=1$ if $x>0$, and zero otherwise). The ratio P/Q is denoted by ρ and it is referred to as the 'vigilance parameter' of the AART1-NN architecture (for more details about the role of the vigilance parameter, see Carpenter and Grossberg 1987). The output activity $f_r(x_r)$ generated by the reset node is given by the following rule:

$$f_r(x_r) = \begin{cases} 1 & \text{if } x_r > \delta_r \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

Once more, in (13), δ_r represents the threshold of the reset node v_r. In the present model, the value of the bottom-up LTM trace, z_{ij}, and the top-down LTM trace z_{ji} are determined by the following differential equations:

$$\varepsilon_z \frac{d}{dt} z_{ij} = -\left[(L-1)f_1(x_i) + \sum_{k=1}^{M} f_1(x_k)\right] f_2(x_j) z_{ij} + L f_1(x_i) f_2(x_j) \tag{14}$$

$$\varepsilon_z \frac{d}{dt} z_{ij} = f_2(x_j) z_{ij} + f_1(x_i) f_2(x_j) \tag{15}$$

In (14), L is a constant. The parameters ε_1, ε_2, ε_r and ε_z that appear in the previous equations are referred to as 'learning rates'. These values determine the rate of change for the variables (STM activities in the F_1 and F_2 fields, activity of the reset node, and bottom-up or top-down LTM traces) that are characterized by the above equations. A smaller value for the parameter ε results in a faster rate of change for the variable described by the differential equation under consideration. In particular, in the AART1-NN model the rate of change for the STM activities in the F_1 and F_2 fields, as well as the rate of change for the activity of the reset node, are much faster than the rate of change for the bottom-up and top-down LTM traces (i.e. ε_1, ε_2, $\varepsilon_r \ll \varepsilon_z$).

3. Circuit design and implementation

In this section we design an analogue circuit prototype of the AART1-NN, consisting of four nodes in the F_1 field (nodes v_1 through v_4), a reset node v_r, and eight nodes in the F_2 field (nodes v_5 through v_8 in the first layer and nodes $\hat{v}_8$ in the second layer). The analogue circuit implementation of the prototype AART1-NN is based on the set of differential equations described in the previous section. In these equations, the constants (e.g. A_1, A_2), the initial bottom-up and top-down LTM traces (e.g. $z_{15}(0)$, $z_{51}(0)$), and the initial STM activities (e.g. $x_1(0)$, $x_2(0)$) are chosen as depicted in the Table (for why the parameters are chosen as such, see Heileman *et al.* 1992). It is worth pointing out that there is a wide range of parameter values for which the differential equations of § 2 can be implemented (again, for more details see Heileman *et al.* 1992).

The implementation of the prototype circuit is accomplished in a number of steps. In the first step, the circuit cells, such as the weighted summer, comparator, multiplier and integrator, are implemented using operational amplifiers, and their

$A_1=1$	$B_1=1{\cdot}5$	$C_1=1$	$D_1=1$	$\delta_1=0{\cdot}01$
$\varepsilon_1=1$	$A_2=0{\cdot}3$	$B_2=10^4$	$C_2=10^4$	$D_2=1$
$\delta_2=0{\cdot}01$	$\varepsilon_2=100$	$A_r=2$	$P=0{\cdot}99$	$Q=1$
$\delta_r=0{\cdot}02$	$\varepsilon_r=1$	$\delta_2=10^{-4}$	$L=1{\cdot}01$	$\varepsilon_z=10^4$
$z_{15}(0)=0{\cdot}24$	$z_{25}(0)=0{\cdot}24$	$z_{35}(0)=0{\cdot}24$	$z_{45}(0)=0{\cdot}24$	$z_{16}(0)=0{\cdot}22$
$z_{26}(0)=0{\cdot}22$	$z_{36}(0)=0{\cdot}22$	$z_{46}(0)=0{\cdot}22$	$z_{17}(0)=0{\cdot}20$	$z_{27}(0)=0{\cdot}20$
$z_{37}(0)=0{\cdot}20$	$z_{47}(0)=0{\cdot}20$	$z_{18}(0)=0{\cdot}18$	$z_{28}(0)=0{\cdot}18$	$z_{38}(0)=0{\cdot}18$
$z_{48}(0)=0{\cdot}24$	$z_{51}(0)=1$	$z_{52}(0)=1$	$z_{53}(0)=1$	$z_{54}(0)=1$
$z_{61}(0)=1$	$z_{62}(0)=1$	$z_{63}(0)=1$	$z_{64}(0)=1$	$z_{71}(0)=1$
$z_{72}(0)=1$	$z_{73}(0)=1$	$z_{74}(0)=1$	$z_{81}(0)=1$	$z_{82}(0)=1$
$z_{83}(0)=1$	$z_{84}(0)=1$	$x_1(0)=0$	$x_2(0)=0$	$x_3(0)=0$
$x_4(0)=0$	$x_5(0)=0$	$x_6(0)=0$	$x_7(0)=0$	$x_8(0)=0$
$\hat{x}_5(0)=0$	$\hat{x}_6(0)=0$	$\hat{x}_7(0)=0$	$\hat{x}_8(0)=0$	$x_r(0)=0$

Parameter values and initial conditions used for the PSpice/numerical simulation of the differential equations in §2. These differential equations describe the prototype AART1-NN designed in §3.

functionality is tested. In the second step, we implement an arbitrary node belonging to the first layer of the F_2 field. In the third step, we design an arbitrary node belonging to the F_1 field. Then, in the fourth step, we proceeded with the implementation of the reset node and the nodes in the second layer of the F_2 field. Finally, in the fifth step, the bottom-up and top-down LTM traces were implemented to interconnect the nodes in the F_1 and the F_2 field. Each of the above implemented circuits was tested extensively to verify its functionality. The functionality of the entire prototype AART1-NN was also tested and the details of the testing procedure are included in the following sections.

Let us now describe in more detail the circuit implementation corresponding to an arbitrary node in the first layer of the F_2 field (i.e. the second step). In the AART1-NN, the activity x_j of the node v_j in the first layer of the F_2 field is described by (5) (in our case $M=4$ and $N=8$). Equation (5) can be rewritten as follows:

$$\varepsilon_2 \frac{\mathrm{d}}{\mathrm{d}t} x_j = -(1 + A_2 J_j^+ + C_2 J_j^-) x_j + (J_j^+ - B_2 J_j^-) \tag{16}$$

where

$$\left.\begin{aligned} J_j^+ &= f_2(x_j) g(I) + D_2 \sum_{i=1}^{4} f_1(x_i) z_{ij} \\ J_j^+ &= \sum_{k \neq j} f_2(x_j) + \hat{f}_2(\hat{x}_j) \end{aligned}\right\} \tag{17}$$

The value of the function $g(I)$ in the above equations is determined by the summation of the binary inputs affecting the F_1 field of the AART1-NN. The quantities $f_1(x_i)$ and $\hat{f}_2(\hat{x}_j)$ correspond to output activities of nodes v_i and $\hat{v}_j$, respectively; the activities of these nodes are determined by the differential equations (1) and (11), which are implemented in Steps 3 and 4, respectively. Finally, z_{ij} is the bottom-up LTM trace from node v_i to node v_j determined by (14), which is

implemented in Step 5. The circuit implementation of the differential equation corresponding to the activity x_j (see (16)) can be realized by a weighted summer with inputs x_j, $A_2 J_j^+ x_j$, $C_2 J_j^- x_j$, $B_2 J_j^-$ and $-J_j^+$; the output of the weighted summer would be $\varepsilon_2(\mathrm{d}/\mathrm{d}t)x_j$. The relationship between x_j and $\varepsilon_2(\mathrm{d}/\mathrm{d}t)x_j$ is accomplished by utilizing an integrator combined with an inverter. The circuit diagram for node v_j in the first layer of the F_2 field is shown in Fig. 2. In this figure the triangles represent the operational amplifiers, and blocks A, B, C, D and E denote the inverter, weighted summer, integrator, multiplier, and comparator cells, respectively. As we mentioned above (first step), each cell circuit was verified using PSpice simulation before it was incorporated into the one-node circuit of Fig. 2. Note that the value of the learning rate ε_2 is determined by the RC time constant of the integrator (see block C in Fig. 2). Furthermore, the threshold function $f_2(x_j)$, described in (4), is provided by the comparator with input x_j (see Block E in Fig. 2). Multipliers are used to implement the multiplication items on the right hand side of (16), such as $A_2 J_j^+ x_j$, $C_2 J_j^- x_j$, $B_2 J_j^-$, and $-J_j^+$.

Figure 3 shows the results obtained from the PSpice simulation of the differential equation (16) using a near ideal op.-amp. model (henceforth, this model is referred to as the ideal op.-amp. model), and using a practical op.-amp. 741 model. In Fig. 3, we have implemented the differential equation (16) for an arbitrary node v_j in the first layer of the F_2 field, with $A_2 = 0{\cdot}3$, $B_2 = 100$, $C_2 = 100$, $D_2 = 1$, $\delta_2 = 0{\cdot}5$, $\varepsilon_2 = 1$, $g(I) = 1$ and $\sum_{i=1}^{4} f_1(x_i) z_{ij} = 1$. As we can see from Fig. 3, excellent agreement is found between exact solutions and the PSpice simulation results using the ideal op.-amp. model. The ideal op.-amp. amplifier model used as a very large input resistance (100 kΩ), a very small output resistance (10 Ω), and a high voltage gain (10^6). Several practical op.-amp. models are available in the model library of the PSpice simulator, each one with characteristics resembling the characteristics of a practical op.-amp. commercially available. We chose the op.-amp. 741 model because it has been widely used in circuit applications such as signal processing and power amplification. As Fig. 3 demonstrates, the simulation results obtained by using the op.-amp. 741 model follow the exact trend of the simulation results produced by using the ideal op.-amp. model. The only difference is a small propagation delay caused by the RC time constant associated with the real op.-amp. This delay is not going to affect the behaviour of the prototype AART1-NN that we intend to build. The only effect that it might have is to introduce an additional delay for the time required by the nodes in the AART1-NN to become active or non-active. Thus, in order to reduce the CPU time for the PSpice simulation, we employ the ideal op.-amp. model for all our circuit designs.

Next, the circuit for an arbitrary node v_i in the F_1 field is designed, based on (1). Following a procedure similar to that described above, the circuit is implemented as shown in Fig. 4. The output of this circuit (i.e. x_i) is used to produce the output activity $f_1(x_i)$ of node v_i in the F_1 field, which affects the nodes v_j in the first layer of the F_2 field (see Fig. 2). Subsequently, the differential equations that determine the activity of an arbitrary node $\hat{v}_j$ in the second layer of the F_2 field (see (11)), and the activity of the reset node (see (12)) are implemented; the corresponding circuits are shown in Figs 5(*a*) and 5(*b*). Note that the circuit corresponding to $\hat{x}_j$ generates the output activity signal, $\hat{f}(\hat{x}_j)$, of node $\hat{v}_j$, which affects node v_j in the first layer of the F_2 field (see Fig. 2). In addition to that, the circuit corresponding to x_r generates an output activity $f_r(x_r)$ which influences all the nodes in the second layer of the F_2 field

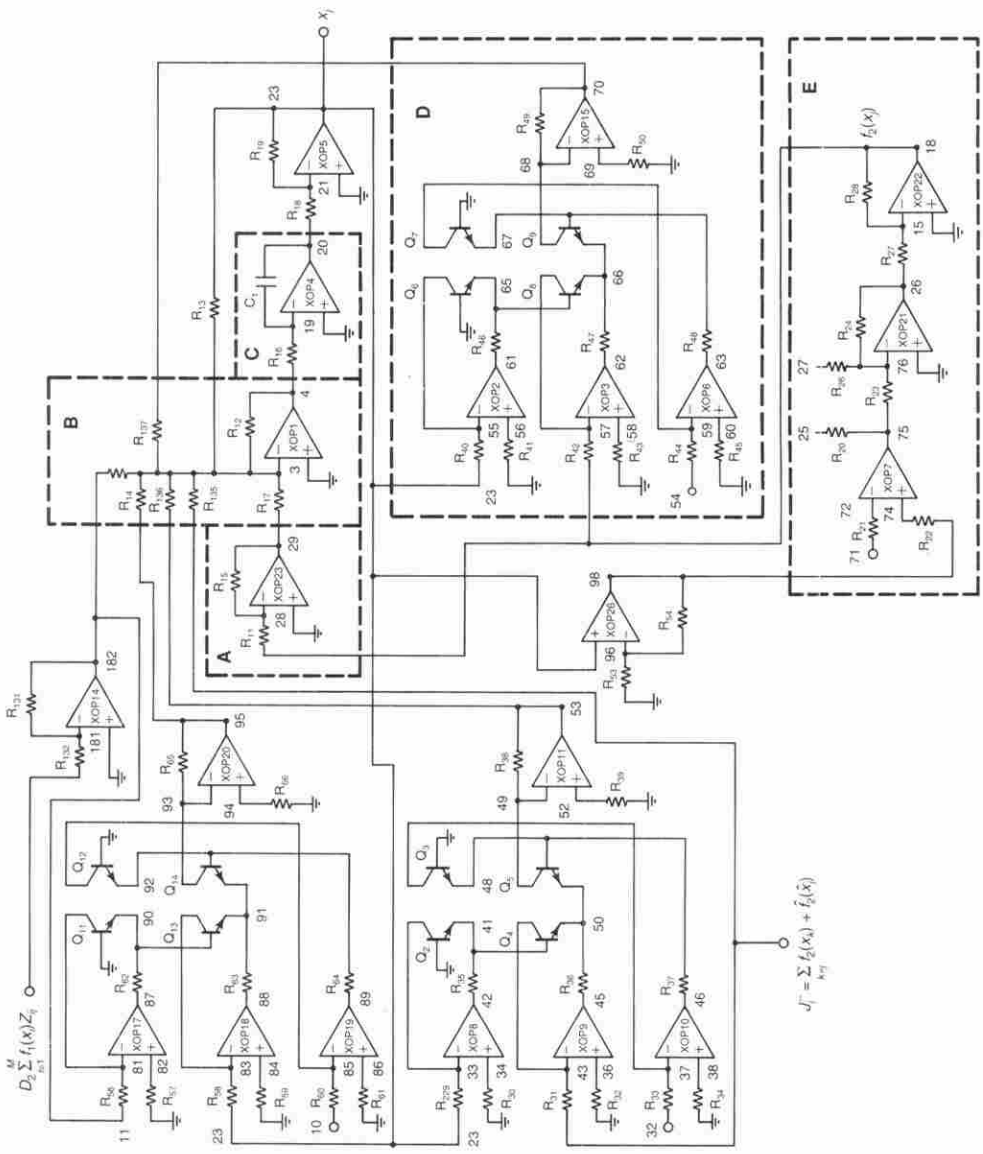


Figure 2. The circuit diagram for an arbitrary node v_j in the first layer of the F_2 field. Blocks A, B, C, D and E are circuit cells for the inverter, weighted summer, integrator, multiplier, and comparator, respectively.

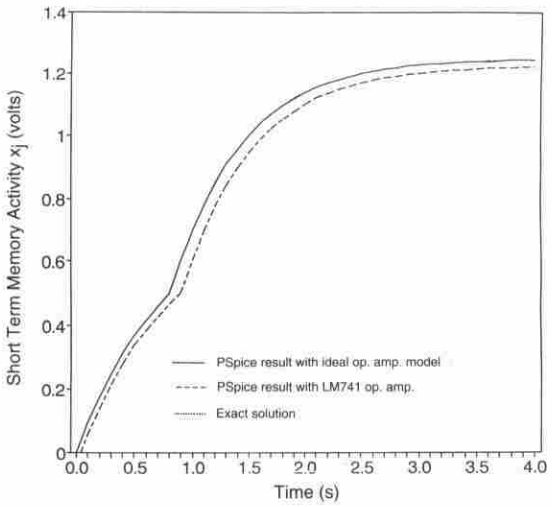


Figure 3. Short term memory (STM) activity of an arbitrary node v_j in the first layer of the F_2 field using an ideal op.-amp. and LM741 op.-amp.

(see Fig. 5(*a*)). The bottom-up and top-down LTM traces are characterized by (14) and (15); their implemented circuits are shown in Figs 6(*a*) and 6(*b*), respectively.

4. Circuit functionality

In the previous section we designed a circuit prototype of the AART1-NN, consisting of four nodes in the F_1 field (nodes v_1 through v_4), a reset node (node v_r),

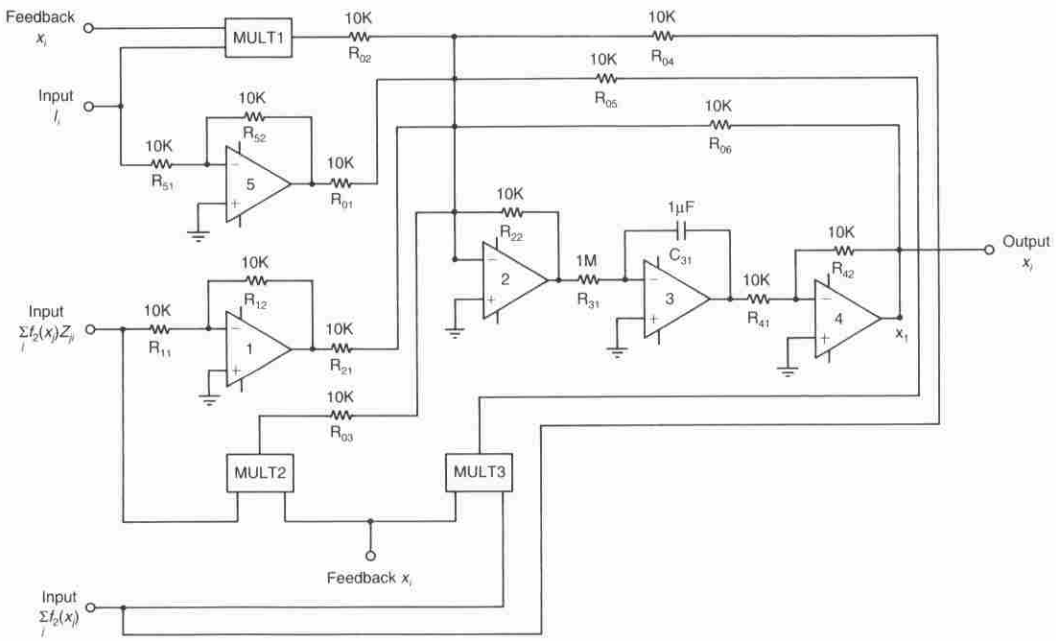


Figure 4. The circuit diagram for an arbitrary node v_i in the F_1 field. The circuit cells depicted in the diagram are designed as demonstrated in Fig. 2.

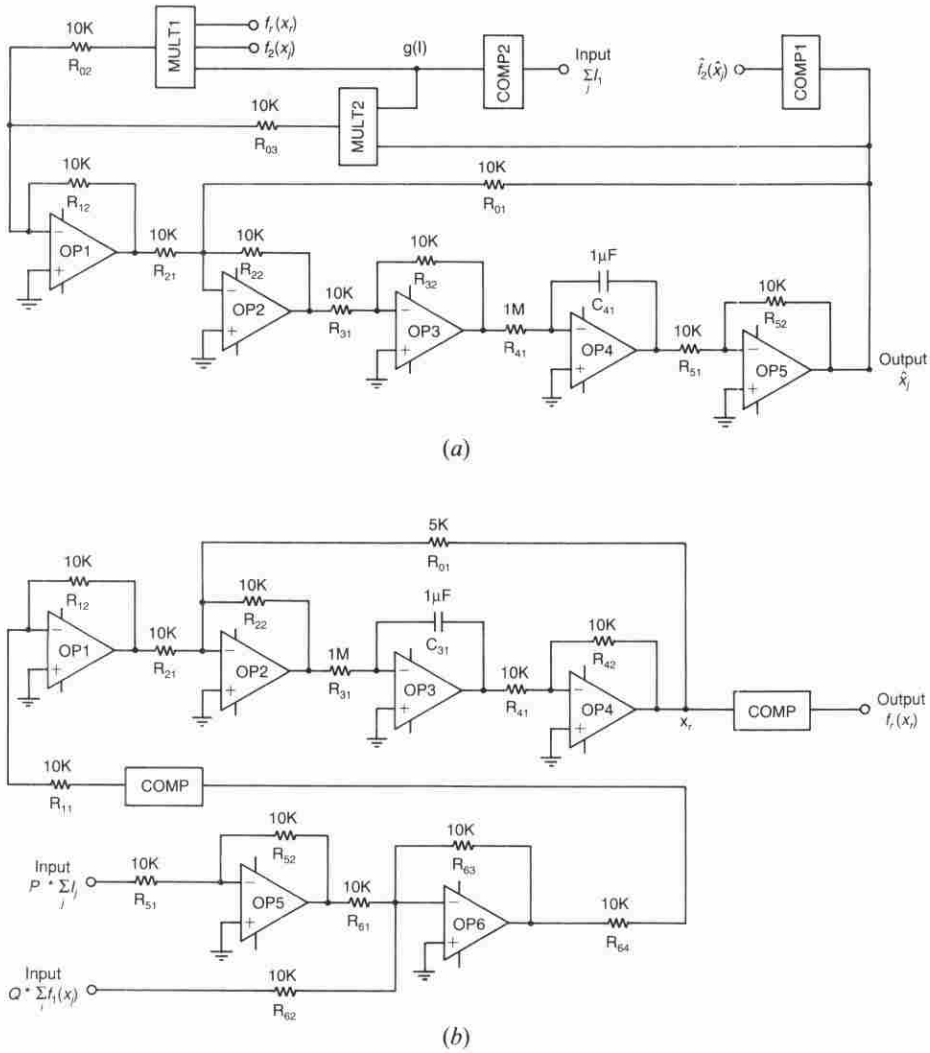


Figure 5. (*a*) Circuit diagram for an arbitrary node $\hat{v}_j$ in the second layer of the F_2 field; (*b*) circuit diagram for the reset node v_r. The circuit cells depicted in the diagrams are designed as demonstrated in Fig. 2.

and eight nodes in the F_2 field (nodes v_5 through v_8 in the first layer and nodes $\hat{v}_5$ through $\hat{v}_8$ in the second layer). In this section, we qualitatively discuss the functionality of the circuit during the presentation of the patterns 1000 and 1100 at the F_1 field. We choose the vigilance parameter $\rho(=P/Q)$ in the network approximately equal to one (see the Table); consequently, we expect the input patterns 1000 and 1100 to choose different categories in the first layer of the F_2 field. The successful operation of the AART1-NN and other ART architectures are based on the assumption that the STM network activity returns to zero between the presentations of two distinct input patterns. In the AART1-NN model, to guarantee this assumption we need to present the zero pattern at the F_1 field between any two distinct pattern presentations, and we need to choose the initial STM activities equal

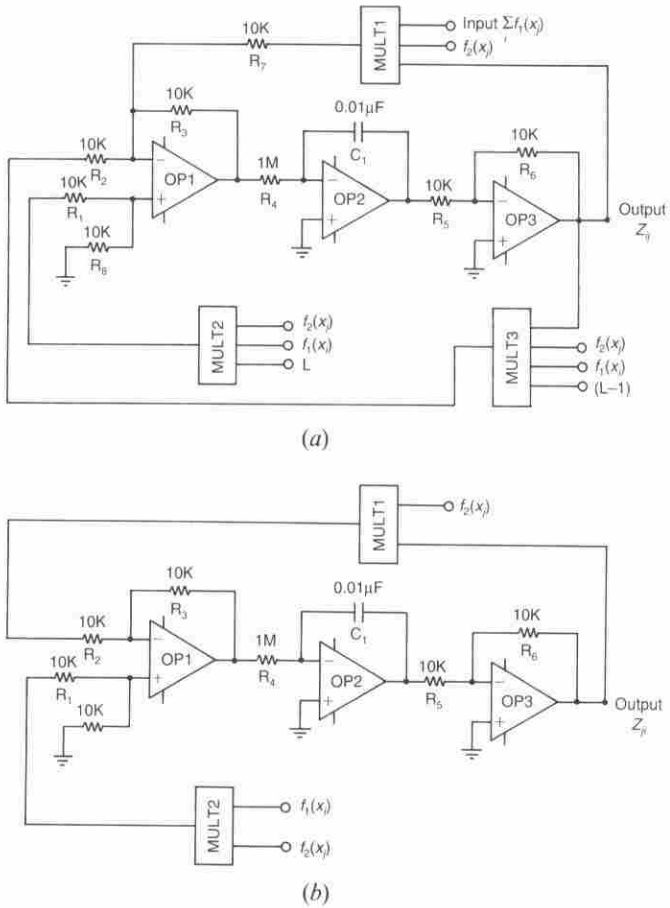


Figure 6. (*a*) Circuit diagram corresponding to an arbitrary bottom-up LTM trace z_{ij}; (*b*) circuit diagram corresponding to an arbitrary top-down LTM trace z_{ji}. The circuit cells depicted in the diagrams are designed as demonstrated in Fig. 2.

to zero (see the Table). Hence for our example we need to present the zero pattern (i.e. pattern 0000) between the presentation of patterns 1000 and 1100. The time during which the zero pattern is presented is much shorter than the time during which useful patterns (such as 1000 and 1100) are presented. We designate the first useful pattern presented to the AART1-NN as I^1 (i.e. $I^1=1000$), and the second useful pattern presented to the network as I^3 (i.e., $I^3=1100$); we reserve the notation I^2 for the zero pattern presented between the I^1 and the I^3 patterns (i.e. $I^2=0000$). We also designate by I_i^1, I_i^2 and I_i^3 ($1\leqslant i\leqslant 4$), the ith component of patterns I^1, I^2 and I^3, respectively. For example, during the presentation of pattern $I^1=1000$ at the F_1 field, node v_1 receives input $I_1^1=1$, node v_2 receives input $I_2^1=0$, node v_3 receives input $I_3^1=0$, and node v_4 receives input $I_4^1=0$. As mentioned in the previous section, the parameters of the differential equations in § 2 and the initial bottom-up and top-down LTM traces during the circuit simulation are chosen as depicted in the Table.

The AART1-NN is supposed to work as follows during the presentation of the patterns I^1, I^2 and I^3. After the appearance of pattern I^1 at the F_1 field, node v_5 is

chosen in the first layer of the F_2 field to represent the input pattern. This is because node v_5 is the node that receives the largest bottom-up input from F_1. Since node v_5 has not been chosen to represent any pattern prior to the appearance of I^1, there is not going to be a mismatch between bottom-up and top-down inputs at the F_1 field; consequently, the reset node will remain non-active, and node v_5 will be deemed approxpriate by the architecture to represent pattern I^1. Subsequently, the bottom-up and top-down traces corresponding to node v_5 will learn the input pattern I^1. After the withdrawal of pattern I^1 from the F_1 field, the zero pattern I^2 is presented to prepare the network for the appearance of the next useful pattern I^3. During the presentation of pattern I^2 at the F_1 field, all STM node activities return to their resting value of zero. When pattern I^3 is presented to the AART1-NN, node v_5 in the first layer of the F_2 field will be chosen first to represent the input pattern I^3. This is due to the fact that node v_5 will be the node receiving the largest bottom-up input from the F_1 field. Node v_5 will be reset by the reset node because there is a mismatch between the bottom-up and top-down inputs at the F_1 field. The bottom-up inputs at the F_1 field correspond to the input pattern I^3 (1100), while the top-down inputs from node v_5 correspond to the pattern I^1 (1000) that node v_5 learned during the presentation of I^1 at the F_1 field. Since the vigilance parameter ρ was chosen approximately equal to one, an almost perfect match between the bottom-up and top-down inputs at the F_1 field is required to keep the reset node non-activated. The reset of node v_5 will force the F_2 field to choose another node in its category representation layer to represent the input pattern I^3. At this time, node v_6 will be chosen since it is the node in the first layer of the F_2 field that receives the next (to node v_5) largest bottom-up input from the F_1 field. The activation of node v_6 will not create a mismatch of bottom-up and top-down inputs at the F_1 field because node v_6 has not been chosen to represent any pattern prior to the appearance of pattern I^3. Hence, the reset node remains non-active and node v_6 will learn to represent the input pattern I^3 (i.e. the bottom-up and top-down LTM traces corresponding to node v_6 will converge to appropriate values).

5. Simulation results

In this section we verify the prototype AART1-NN circuit of § 3 using the PSpice circuit simulator. The PSpice simulation results for the transient response of the prototype circuit are shown in Figs 7–15 (solid curves). Pattern I^1 is presented in the interval between 0 and 70×10^3 s, pattern I^2 is presented in the interval between 70×10^3 and 75×10^3 s, and pattern I^3 is presented in the interval between 75×10^3 and 145×10^3 s. The qualitative verification of our PSpice simulation results is presented below, and it is based on the expected network behaviour, outlined in the previous section. The quantitative verification of the PSpice simulation results is also provided in Figs 7–15 (broken curves) by solving the differential equations of § 2 numerically. The differential equations were solved numercially bu using the Runge-Kutta technique with a step size of 10^{-2}.

In Fig. 7, we demonstrate that node v_5 is the one chosen to represent the input pattern I^1. As we can see from Fig. 7, after the input pattern is presented the activity of v_1 increases from zero to a positive value above the node threshold δ_1. Once v_1 becomes active, nodes in the first layer of the F_2 field begin to receive bottom-up input (i.e. input from the F_1 field). Since v_5 receives the largest bottom-up input (see the initial bottom-up traces in the Table), it becomes active before any other node

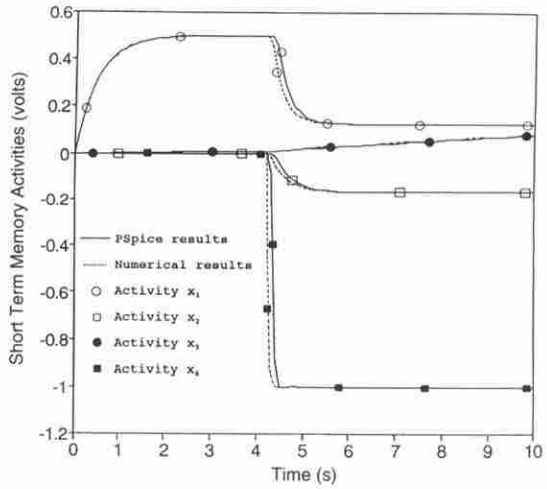


Figure 7. Short term memory (STM) activities in the AART1-NN prototype during the presentation of the $I^1 = 1000$ input pattern. The solid curves correspond to the PSpice simulation results and the broken curves correspond to the numerical simulation results; x_1, x_2, x_5 and x_6 are the STM activities of nodes v_1, v_2 (F_1 field), v_5 and v_6 (F_2 field), respectively. The horizontal axis represents Time in seconds.

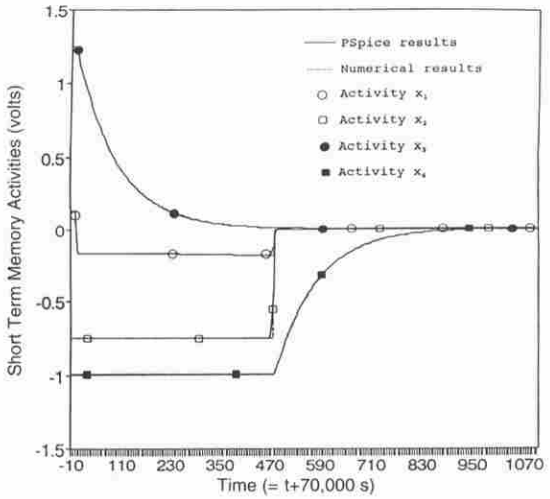
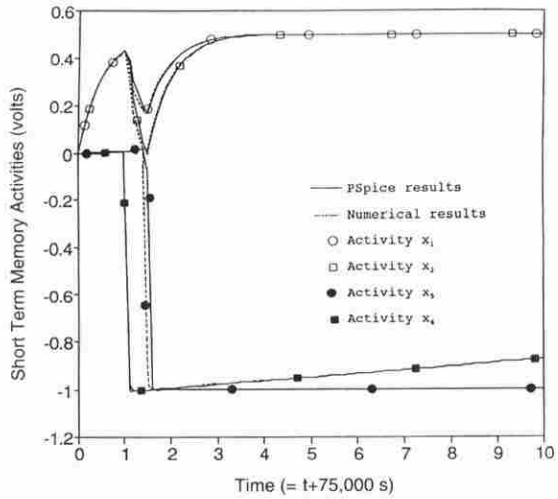
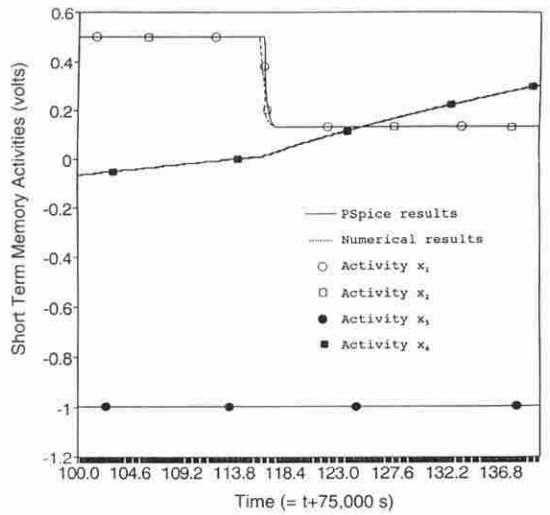


Figure 8. Short term memory activities in the AART1-NN prototype during the presentation of the $I^2 = 0000$ input pattern. The solid curves correspond to the PSpice simulation results and the broken curves correspond to the numercial simulation results; x_1, x_2, x_5 and x_6 are the STM activities of nodes v_1, v_2 (F_1 field), v_5, and v_6 (F_2 field), respectively. The horizontal axis represents Time, where Time is equal to $(t + 70\,000)$ s, and the values of t are shown below the horizontal axis.



(*a*)



(*b*)

Figure 9. Short term memory activities in the AART1-NN prototype during the presentation of the $I^3 = 1100$ input pattern. The solid curves correspond to the PSpice simulation results and the broken curves correspond to the numerical simulation results; x_1, x_2, x_5, x_6 and x_r are the STM activities of the nodes v_1, v_2 (F_1 field), v_5, v_6 (F_2 field) and the reset node v_r, respectively: (*a*) STM activities for the time interval [75 000, 75 010]; (*b*) STM activities for the time interval [75 100, 75 140]. The horizontal axes in both figures represent Time, where Time is equal to $(t + 75\,000)$ s, and the values of t are shown below the horizontal axes.

becomes active in the first layer of the F_2 field. As we can see from Fig. 7, the activity of v_6 which is the node receiving the next largest input, never manages to reach a level above the node threshold δ_2; similar behaviour is exhibited by the other nodes in the first layer of the F_2 field (i.e. nodes v_7 and v_8). After the activation of node v_5, node v_1 in the F_1 field receives both bottom-up input (from the input pattern I^1) and

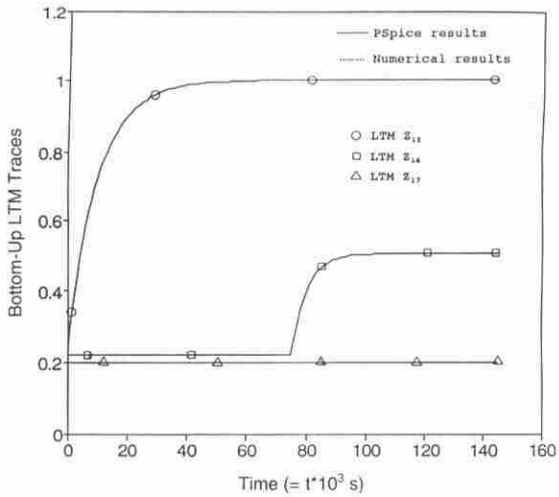


Figure 10. The behaviour of the bottom-up LTM traces z_{15}, z_{16} and z_{17} during the presentation of patterns I^1, I^2 and I^3 at the F_1 field (time interval [0, 145 000]). The solid curves correspond to the PSpice simulation results, while the broken curves coresond to the numerical simulation results. In most instances, the solid and the broken curves overlap. The horizontal axis represents Time, where Time is equal to $(t \times 1000)$ s, and the values of t are shown below the horizontal axis.

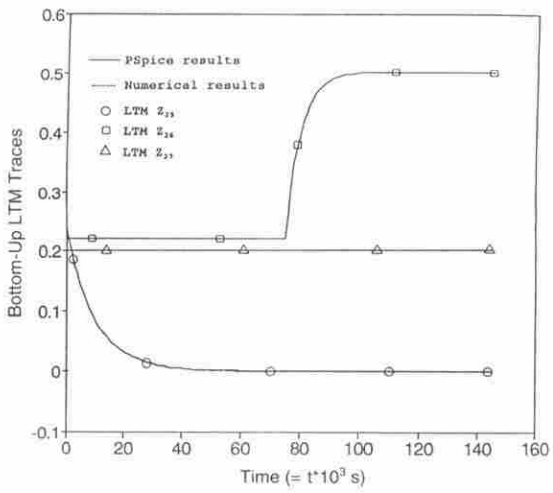


Figure 11. The behaviour of the bottom-up LTM traces z_{25}, z_{26} and z_{27} during the presentation of patterns I^1, I^2 and I^3 at the F_1 field (time interval [0, 145 000]) The solid curves correspond to the PSpice simulation results, while the broken curves correspond to the numerical simulation results. In most instances, the solid and the broken curves overlap. The horizontal axis represents Time, where Time is equal to $(t \times 1000)$ s, and the values of t are shown below the horizontal axis.

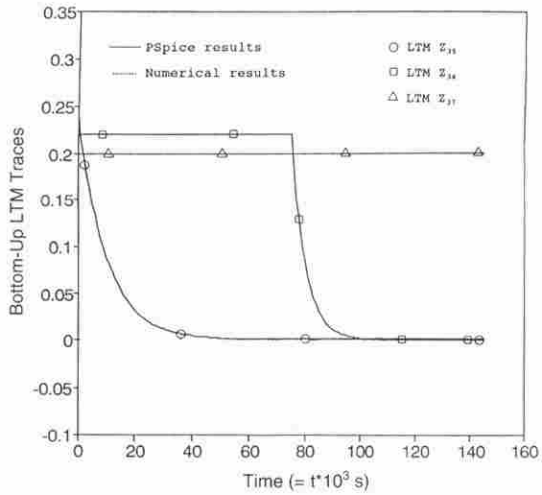


Figure 12. The behaviour of the bottom-up LTM traces z_{35}, z_{36} and z_{37} during the presentation of patterns I^1, I^2 and I^3 at the F_1 field (time interval [0, 145 000]) The solid curves correspond to the PSpice simulation results, while the broken curves correspond to the numerical simulation results. In most instances, the solid and the broken curves overlap. The horizontal axis represents Time, where Time is equal to $(t \times 1000)$ s, and the values of t are shown below the horizontal axis.

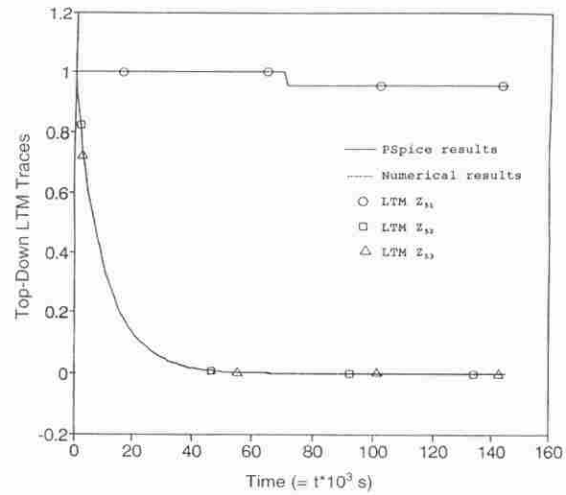


Figure 13. The behaviour of the top-down LTM traces z_{51}, z_{52} and z_{53} during the presentation of patterns I^1, I^2 and I^3 at the F_1 field (time interval [0, 145 000]) The solid curves correspond to the PSpice simulation results, while the broken curves correspond to the numerical simulation results. In most instances, the solid and the broken curves overlap. The horizontal axis represents Time, where Time is equal to $(t \times 1000)$ s, and the values of t are shown below the horizontal axis.

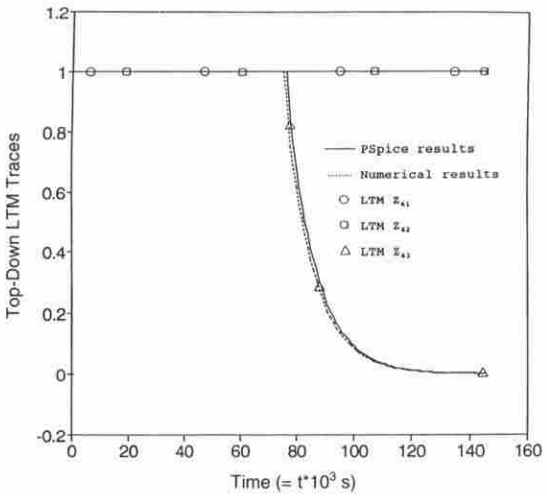


Figure 14. The behaviour of the top-down LTM traces z_{61}, z_{62} and z_{63} during the presentation of patterns I^1, I^2 and I^3 at the F_1 field (time interval [0, 145 000]) The solid curves correspond to the PSpice simulation results, while the broken curves correspond to the numerical simulation results. In most instances, the solid and the broken curves overlap. The horizontal axis represents Time, where Time is equal to $(t \times 1000)$ s, and the values of t are shown below the horizontal axis.

top-down input (from node v_5). This causes the activity of v_1 to decrease and eventually reach a limiting value, which is above the node threshold δ_1(see Fig. 7). Note that the activity of field F_1 throughout the presentation of I^1 is large enough so that the reset node cannot generate a reset wave. Consequently, node v_5 is deemed by the architecture as the right node in F_2 to represent the input pattern I^1, and the

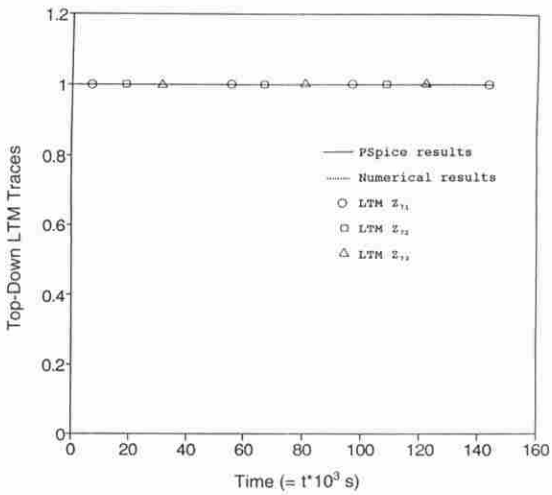


Figure 15. The behaviour of the top-down LTM traces z_{71}, z_{72} and z_{73} during the presentation of patterns I^1, I^2 and I^3 at the F_1 field (time interval [0, 145 000]) The solid curves correspond to the PSpice simulation results, while the broken curves correspond to the numerical simulation results. At most instances, the solid and the broken curves overlap. The horizontal axis represents Time, where Time is equal to $(t \times 1000)$ s, and the values of t are shown below the horizontal axis.

bottom-up and top-down LTM traces converging to node v_5 will learn pattern I^1. In particular, as we can see from Figs 10–13, the LTM traces z_{51} and z_{15} converge to large values, while the remaining traces corresponding to node v_5 converge to small values (i.e. values close to zero).

In Fig. 8, we illustrate what happens during the presentation of the zero pattern at the F_1 field of the AART1-NN. As shown in Fig. 8, all STM node activities converge to their resting value of zero within the first 1000 units of time after the appearance of I^2 (actually, in Fig. 8 only certain STM node activities are shown in order to preserve the clarity of the figure). The behaviour of the network after the presentation of I^3 is depicted in Figs 9(*a*) and 9(*b*). In Fig. 9(*a*), we can see that after the presentation of I^3 node v_5 becomes active first because it is the node in the first layer of the F_2 field that receives the largest bottom-up input. Once node v_5 becomes active, the activities of nodes v_1 and v_2 begin to decrease. The activity of v_1 remains above the threshold, while the activity of vc_2 decreases to a level below the threshold. This a consequence of the fact that node v_1 receives strong top-down input (z_{51} large from Fig. 13), while node v_2 receives weak top-down input (z_{42} is equal to zero from Fig. 13)—both nodes receive bottom-up input. When v_2 becomes non-active, the activity of the reset node starts increasing due to the mismatch between the bottom-up and top-down inputs that is now occurring at the F_1 field. When v_r becomes active (i.e. its activity exceeds the threshold δ_r), it generates a reset wave that deactivates v_5 almost instantaneously (see Fig. 9(*a*)). After v_5 is deactivated, v_1 and v_2 receive only bottom-up input, and their activities increase towards the limiting value of 0·5 (see Fig. 9(*a*)). Now that v_5 is deactivated, v_6 becomes active next since it is the node in the first layer of the F_2 field that receives the next largest bottom-up input from the F_1 field. The activation of v_6 is shown in Figs 9(*a*) and 9(*b*). When v_6 becomes active, the activities of v_1 and v_2 begin to decrease from the value of 0·5; but they both remain above the threshold δ_1 (see Fig. 9(*b*)). This is a consequence of both v_1 and v_2 receiving bottom-up input and strong top-down input (see Fig. 14). It is worth noting that the activity of the reset node starts decreasing some time after the deactivation of node v_5 and that it continues to do so after the activation of v_6 because there is no mismatch between bottom-up and top-down inputs at the F_1 field when node v_6 is active (the activity of the reset node is not shown in Figs 9(*a*) and 9(*b*) in order to preserve the figure clarity). As we emphasized before, node v_6 should not be reset since it has not been chosen to represent any input pattern yet. The input pattern $I^3 = 1100$ is withdrawn from the F_1 field at time 145 000. During the time interval from the activation of v_6 until the withdrawal of pattern I^3 from the F_1 field, the bottom-up and top-down LTM traces corresponding to node v_6 learn to represent pattern I^3. In particular, as can be seen from Figs 10–12 and 14, the LTM traces z_{16}, z_{26}, z_{61} and z_{62} converge to large values, while the remaining traces corresponding to node v_6 converge to small values (i.e. values close to zero). It is worth pointing out that LTM traces, corresponding to nodes that have not been chosen to represent any pattern yet, do not change their values during the presentation and learning of the useful patterns I^1 and I^3 (see the bottom-up or top-down LTM traces corresponding to node v_7 in Figs 10, 11, 12 and 15).

6. Conclusions

In this paper, a prototype circuit has been successfully designed and implemented for the augmented adaptive resonance theory—1 neural network (AART1-NN). The AART1-NN, as its predecessor the ART1-NN, can cluster in a parallel manner an

arbitrary collection of binary input patterns. This capability makes the AART1-NN very attractive for high speed signal processing applications. Furthermore, using the AART1-NN module other adaptive resonance theory (ART) neural networks, which can function as pattern classifiers, can be implemented. The circuit was designed and implemented based on analogue electronic components including operational amplifiers, transistors, capacitors and resistors. Verification of the circuit was carried out using the PSpice circuit simulator available on Sun workstations. The CPU time required for a complete PSpice simulation (i.e. presentation of I^1 and I^3 for 70×10^3 s and presentation of I^2 for 5×10^3 s) ranged between 20 min and 4 h, depending on the values chosen for the learning rates (i.e. the ε's). Excellent agreement was obtained when we compared the PSpice simulation results and the results calculated (either exactly or numerically) from the coupled differential equations constituting the neural network. The following features are important regarding the work performed.

(1) The implemented circuit exhibits the same behaviour as the ART1-NN. The ART1-NN has been used successfully in numerous applications so far.

(2) The implemented circuit covers all the subsystems of the ART1-NN architecture, such as the attentional subsystem (i.e. F_1 and F_2 fields), and the orienting subsystem. As a result, the circuit developed is self-contained and does not require any external supervision. The only information required in order for the circuit to function properly, is the collection of the input patterns affecting the F_1 field. This is a feature that distinguishes our implementation from other implementations of the ART1-NN reported in the literature.

(3) The implemented circuit is parallel and analogue. The parallel characteristic allows the circuit to possess very fast processing capabilities.

One of our future goals is to extend the designed AART1-NN prototype circuit to larger and consequently more practical prototypes, as well as utilizing the aforementioned design approach and implementation for other similar neural network architectures. Finally, another worthwhile pursuit, which will make the circuit more compact and integrable into one chip, is to implement the AART1-NN based on the approach of very large scale integration using only CMOS technology.

Acknowledgment

This work was supported in part by the Florida High Technology Council, by the Division of Sponsored Research at UCF, and by the Harris Semiconductor Corp., Melbourne, Illinois.

References

Carpenter, G. A., 1989, Neural network models for pattern recognition and associative memory. *Neural Networks*, **2**, 243–257.

Carpenter, G. A., and Grossberg, S., 1987, A massive parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics and Image Processing*, **37**, 54–115.

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J. H., and Rosen, D. B., 1992, Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, **3**, 698–713.

CARPENTER, G. A., GROSSBERG, S., and REYNOLDS, J. H., 1991, ARTMAP: supervised real-time learning of non-stationary data by a self-organizing neural network. *Neural Networks*, **4,** 565–588.

CAUDELL, T. P., 1991, Adaptive neural systems. 1990 IR & D Technical Report, TR # BCS-CS-ACS-91-001, Research and Technology Boeing Computer Services, Seattle, Washington, U.S.A.

FRYRE, R. C., RIETMAN, E. A., and WONG, C. C., 1991, Back-propagation learning and non-idealities in analog-neural network hardware. *IEEE Transactions on Neural Networks*, **2,** 110–117.

HEILEMAN, G. L., GEORGIOPOULOS, M., and ABDALLAH, 1992, A dynamical adaptive resonance architecture. *IEEE Transactions on Neural Networks*, submitted for publication; also partially published in the *Proceedings of the International Joint Conference on Neural Networks*, Singapore, November 1991, Vol. 3, pp. 2658–2663.

NAHET, B., DARLING, R. B., and PINTER, R. B., 1989, Analog implementation of shunting neural networks. *Proceedings of the 1988 Neural Information Processing Systems* (Morgan Kaufman), pp. 695–701.

READ, R. D., and GEIGER, R. L., 1989, A multiple-input OTA circuit for neural networks. *IEEE Transportations on Circuits and Systems*, **36,** 767–770.

TSAY, S. W., and NEWCOMB, R., 1991, VLSI Implementation of ART1 memories. *IEEE Transactions on Neural Networks*, **2,** 214–221.