A Heteroassociative Memory Using Current-Mode MOS Analog VLSI Circuits

KWABENA A. BOAHEN, PHILIPPE O. POULIQUEN, ANDREAS G. ANDREOU, MEMBER, IEEE, AND ROBERT E. JENKINS, MEMBER, IEEE

Abstract —**We describe a scalable architecture for the implementation of neural networks that produces regular and dense designs. A combination of low power consumption and enhanced performance is achieved by using analog current-mode MOS circuits operating in subthreshold conduction.**

We have designed and fabricated a bidirectional associative memory in 3-μm bulk CMOS. The chip has 46 neurons arranged in three layers—a hidden layer and two input/output layers. There are 448 repeatedly programmable connections. This chip performs two-way associative search for stored vector pairs and has optimal storage efficiency of one hardware bit per information bit. The synaptic elements have bipolar current outputs. These currents are integrated using the interconnect capacitance to determine the activation of the thresholding neurons. The unit synaptic current I_u is externally programmable. Recall rates of 100 000 vectors per second have been obtained with $I_u = 0.5$ μA.

I. Introduction

Biological information processing systems outperform modern digital machines in problems that require processing large amounts of fuzzy, noisy, real world data, such as pattern recognition and classification. The shortcomings of conventional approaches have forced computer scientists and engineers to borrow paradigms from biology to solve problems in sensory perception and machine intelligence. In addition to handling noisy and even novel inputs, neuromorphic systems have two other desirable features: fault tolerance and massive parallelism.

The *smart memories* project, using an elegant five-transistor memory cell design [1], and work by Jones *et al.* [2] emphasized digital VLSI content addressable memories for specialized computing engines. Also, parallel programming languages, such as Linda [3], use *associative look-up* to create and coordinate processes. However, no digital implementation of an associative processing system can capture the central idea of a *physical* system that is able to store and process information like the brain [4]. Neural paradigms for associative memories have been proposed and investigated in the past [5], [6]. The computational capability of these models has been demonstrated with problems in pattern recognition, vector quantization, novelty filtering, and optimization [5]–[7].

The Hopfield neural model was implemented in VLSI by Sivilotti *et al.* [8]. This was the first successful single-chip implementation of a programmable neural circuit for an associative memory. This chip and subsequent projects showed how digital-oriented MOS VLSI processes can be used to implement large scale analog systems [9], [10]. Power dissipation levels compatible with very large scale integration are achieved by operating the devices in the subthreshold conduction region.

In this paper we present an analog VLSI architecture for associative memories that uses current signals and native device physics to implement area-efficient computational primitives. In the next section we describe the heteroassociative neural network we developed to make optimal use of digital memory. This model is equivalent to Kosko's bidirectional associative memory (BAM) [11] and includes the Hopfield net as a special case. Our model differs in that it has a hidden layer that uses a unary representation to store the vector pairings. As a result, only one-bit weights are needed. We show that this three-layer model has optimal storage efficiency of one hardware bit per information bit.

In Section III we review subthreshold MOSFET behavior and current mirrors, the primary computational elements in current-mode (CM) circuits. Section IV introduces our synthetic neural subcircuits. Simple CM circuits that perform the functions of thresholding neurons, nonthresholding neurons and synapses are described. In the following section (Section IV) we present an architecture that uses these circuits, in a regular structure, to implement the three layer BAM model. By using transistors as coupling elements and circuits with current inputs the problems of fan-in and fan-out are solved in a natural way. As a result, our architecture is scalable. Preliminary test results obtained from fabricated chips are presented.

Manuscript received July 5, 1988; revised November 7, 1988 and December 22, 1988. This work was supported by the Independent Research and Development Program of the Johns Hopkins University, Applied Physics Laboratory, Baltimore, MD. This paper was recommended by Guest Editors R. W. Newcomb and N. El-Leithy.

K. A. Boahen, P. O. Pouliquen, and A. G. Andreou are with the Department of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD 21218.

R. E. Jenkins is with the Applied Physics Laboratory, The Johns Hopkins University, Laurel, MD 20707.

IEEE Log Number 8826719.

0098-4094/89/0500-0747$01.00 ©1989 IEEE

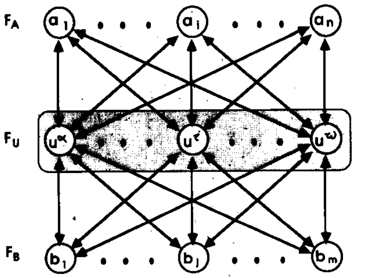


Fig. 1. Three-layer BAM model. A middle-layer neuron (hidden unit) is assigned to each association stored.

II. Associative Memory Models

Let $X = (x_1, x_2, \cdots, x_n)^T$ and $Y = (y_1, y_2, \cdots, y_m)^T$ represent the states of two neuron layers, of size n and m, respectively.[1] Ideally, a *heteroassociative* neural network operates as follows:

In the *store* mode, the current state of each layer is stored, forming the association (A, B).

In the *recall* mode, the network converges to the stored state (A, B) nearest to its initial state (X, Y).

These neurons receive inputs from neurons in another layer through synapses. A neuron's *activation* is the linear sum of these inputs weighted by the synaptic efficacies. We make the following distinctions:

- A *thresholding* neuron has two discrete states, $x = \pm 1$, determined by the sign of its activation, v, that is $x = \text{sgn}(v)$.
- A *non-thresholding* neuron's output equals its activation.

The network we introduce is both bidirectional and symmetric.

- In a *bidirectional* network, neurons in the A field determine the states of those in the B field, and vice versa.
- In a *symmetric* network, if the input to neuron i from neuron j is weighted by w_{ij}, then $w_{ji} = w_{ij}$.

2.1. Three-Layer Bidirectional Associative Memory

This model, shown in Fig. 1, has two input/output layers (F_A and F_B), with n and m thresholding neurons respectively, and a hidden layer (F_U) with s non-thresholding neurons. A similar, but strictly feedforward network, was studied by Baum *et al.* [12]. This network stores up to s associations, labeled by the index set Ω, which are programmed as follows:

A hidden unit is assigned to each association. For association (A^ρ, B^ρ) the weights between the chosen hidden unit (also labeled with the superscript ρ) and neurons in F_A and F_B are simply set to the corresponding component of A^ρ or B^ρ. Thus for F_A; $w_{\rho i} = w_{i\rho} = a_i^\rho$, and for F_B; $w_{\rho j} = w_{j\rho} = b_j^\rho$.

During recall, the states of neurons in either F_A or F_B are initialized. All the neurons are then allowed to update their states by thresholding their activation. We shall show that if the stored vectors satisfy certain conditions the vector pair closest to the initial state is recalled. Formally, if the F_A neurons are initialized to A, then the recalled association (A^σ, B^σ) has the property

$$A^T A^\sigma = \max_{\rho \in \Omega} A^T A^\rho.$$

To show this, observe that the activation, v_j, of the jth neuron in F_B is

$$v_j = \sum_{\rho \in \Omega} w_{j\rho} u^\rho = \sum_{\rho \in \Omega} w_{j\rho} \sum_{i=1}^{n} w_{\rho i} a_i.$$

Since $w_{j\rho} = b_j^\rho$ and $w_{\rho i} = a_i^\rho$, we have

$$v_j = \sum_{\rho \in \Omega} b_j^\rho \sum_{i=1}^{n} a_i^\rho a_i = \sum_{\rho \in \Omega} b_j^\rho A^T A^\rho. \tag{1}$$

Rewriting this equation as

$$v_j = b_j^\sigma A^T A^\sigma + \sum_{\rho \in \Omega,\, \rho \neq \sigma} b_j^\rho A^T A^\rho$$

we find that $y_j = \text{sgn}(v_j) = b_j^\sigma$ if

1) The inner-product $A^T A^\sigma$ between the input vector A and the target vector A^σ is positive, and
2) The sum of the inner-products between the input vector and the other stored vectors is less than $A^T A^\sigma$.

Under these conditions the jth neuron's state becomes b_j^σ and the closest vector B^σ is recalled. Feeding B^σ back through the network yields A^σ if the above conditions hold for the B vectors as well. The condition $A^T A^\sigma > 0$ guarantees that the complement of the target vector is not recalled. If these conditions fail to hold, the recalled vector will be a combination of the stored vectors.

2.2. Equivalent Networks

In this section we show that this three layer network is equivalent to Kosko's two-layer BAM [11].

Indeed, a two-layer BAM with n neurons in F_A and m neurons in F_B has an $n \times m$ connection matrix $M(= [m_{ij}])$ which is the sum of outer-products AB^T

$$m_{ij} = \sum_{\rho \in \Omega} a_i^\rho b_j^\rho \tag{2}$$

During recall, the activation of the jth neuron in F_B is

$$v_j = \sum_{i=1}^{n} m_{ij} a_i. \tag{3}$$

Using (2) and reversing the order of summation, we find

$$v_j = \sum_{i=1}^{n} a_i \sum_{\rho \in \Omega} a_i^\rho b_j^\rho = \sum_{\rho \in \Omega} b_j^\rho \sum_{i=1}^{n} a_i a_i^\rho$$

which is (1).

[1] Column vectors will be denoted by capital letters and their components by small letters with appropriate subscripts.

Kosko proved that every real matrix M is a bidirectionally stable associative memory [11]. Therefore, the three-layer BAM also has convergent trajectories for any set of stored vectors.

From (2), it should be obvious that if $A^{\rho} = B^{\rho}$ for all ρ, the connection matrix is symmetric as in a Hopfield net [6] with n neurons. We compare the hardware requirements of these networks, including the Hopfield net in Section 2.3.

2.3. *Efficient Implementation*

Of these three models, the three-layer BAM has the highest storage and computational efficiency, making it the best candidate for VLSI.

An $n \times n$ two-layer BAM has n^2 weights whereas a Hopfield net with the same number of neurons has nearly four times as many weights, $2n(2n-1)$ to be exact. In these matrix memories, the weights have *integer* values, $|m_{ij}| \leqslant s$ (see (2)), where s is the number of associations stored. These weights require $\log_2 s$ bits and a sign bit. On the other hand, an $n \times n$ three-layer BAM has $2n + s$ neurons and $2ns$ *bipolar* weights, each represented by a single bit. Thus the s vector pairs ($2 \times n$ bits each) are stored using the optimal number of bits. Note that, in practice, 2s hidden layer neurons and $4ns$ synapses are required to handle bidirectional information flow (refer to Section IV).

From these expressions, we can compute the number of memory cells required and consequently the storage *inefficiency* (hardware bits per information bit). We use $s = 2n$ for the Hopfield net and $s = n$ for the BAM networks. Thus s is the maximum number of orthogonal vectors that may be stored and recalled correctly. Results for $n = 32$ are shown in Table I. The three-layer BAM uses the least memory cells because it stores one information bit per hardware bit. It should be pointed out that this analysis would be different if the weights could be stored *and* manipulated in *analog* form.

To compare computational requirements, we count how many computing elements are needed to compute activation for each neuron. We assume a computing element (CE) can perform a multiplication and an addition. In other words, each connection in the network is physically realized by a CE. An $n \times n$ two-layer BAM requires n CE's per neuron, and a $2n$ neuron Hopfield net requires $(2n-1)$ CE's, while a three-layer BAM requires s CE's, plus (ns/m) CE's for the hidden layer; a total of $2s$ for $n = m$. Though the three-layer BAM is only half as efficient as the two-layer BAM, it requires no additional circuitry to manipulate the weights. On the contrary, stored binary representations for the weights in the other networks must be adjusted by

$$m'_{ij} = m_{ij} + a_i b_j$$

(see (2)) for each new association (A, B). This demands an extra CE per connection. Clearly, our choice to implement the three-layer BAM was influenced by the lack of a compact nonvolatile analog storage element in VLSI technology.

TABLE I
COMPARISON OF ASSOCIATIVE MEMORY MODELS

	Hopfield Net	*2-Layer BAM*	*3-Layer BAM*
Neurons	64	64	128
Synapses	4032	2048	4096
Memory (Kbits)	28	6	2
Inefficiency	7	3	1
CE's	64	32	64

III. Low-Power CM MOS Circuits

Neuroprocessors require high degrees of connectivity, that is, large fan-in and fan-out. Our architecture uses transconductances as coupling elements to achieve large fan-out. These transconductances are simply MOS transistors. Voltage inputs are applied to the isolated gate of the transistor to obtain low conductance current outputs at the drain. The fan-in problem is solved by using neurons with current inputs and obtaining the sum of all these currents on a single input line.

Although our circuits operate with very small subthreshold currents, we achieve reasonable speeds by keeping voltage swings small. For a given current signal level, both voltage swings and propagation delays are inversely proportional to the input conductance. Thus by taking advantage of the high transconductance of MOS FET's in subthreshold conduction [9], [13] we obtain a good power/speed tradeoff. Dynamic power dissipation and supply noise are reduced as a result of the smaller voltage swings. This eliminates parasitic charging and discharging currents and allows smaller signals to be used, thereby cutting quiescent dissipation. This approach yields relatively fast analog circuits with power dissipation levels compatible with wafer scale integration.

3.1. *Subthreshold MOSFET Operation*

We operate the MOS transistor in the "off" region, characterized by $V_{\text{gs}} < V_{\text{th}}$ for low power dissipation. This is referred to as the weak-inversion or *subthreshold conduction* region. The transfer characteristics (obtained using a testing system developed at Hopkins [14]) are shown in Fig. 2. Notice that the drain current I_{ds} is exponentially dependent on the gate voltage V_{gs} and bulk (local substrate) voltage V_{bs} over nearly six decades. In the saturation region, the drain current is given by

$$I_{d\,\text{sat}} = \left(\frac{W}{L}\right) I_0 e^{(V_{\text{gs}}/\gamma + V_{\text{bs}}/\eta)/U_T}; \qquad V_{\text{ds}} > V_{d\,\text{sat}} \simeq 4U_T. \quad (4)$$

where,

- W, L *effective* channel width and length, respectively,
- I_0 process dependent parameter,
- γ, η measure the ineffectiveness of the gate and substrate potentials in reducing the barrier. The values $\gamma = 1.9$ and $\eta = 3.4$ for the characteristics shown are typical for a digital oriented CMOS process.
- $U_T(= kT/q)$ thermal voltage—26 mV at room temperature.

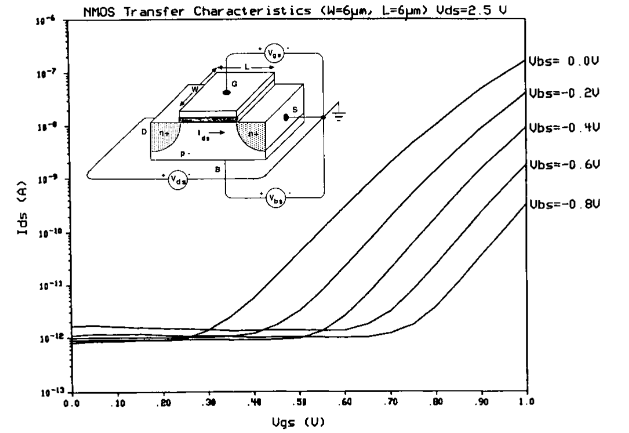
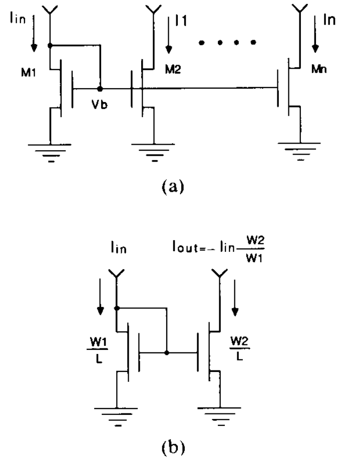


Fig. 2. Subthreshold characteristics for an *N*-type MOS transistor. The variation of the channel current with the substrate voltage is included to point out that the MOS transistor is a four terminal device.



(a)

(b)

Fig. 3. Computation with current mirrors. (a) Replication. (b) Scaling. Although scaling can be achieved by choosing suitable *W* and *L*, this is avoided. Current scaling is accomplished using the appropriate number of equal-size devices in parallel.

The current changes by a decade for a 120 mV change in V_{gs} or a 280 mV change in V_{bs}.

An empirical relationship for the drain conductance is

$$g_{d\,\mathrm{sat}} = \frac{I_{d\,\mathrm{sat}}}{V_0 + V_{ds}} \tag{5}$$

where V_0 is the Early voltage, typically about 55 V. This relation captures the slope of the output characteristic caused by the dependence of L on V_{ds} [15].

From (4) the transconductance is

$$g_m = \frac{\partial I_{d\,\mathrm{sat}}}{\partial V_{gs}} = \frac{I_{d\,\mathrm{sat}}}{\gamma U_T}. \tag{6}$$

These equations sacrifice accuracy for simplicity; they are only meant for rough design calculations. As written, they apply to *n*-type transistors, signs should be reversed to obtain equations for the *p*-type.

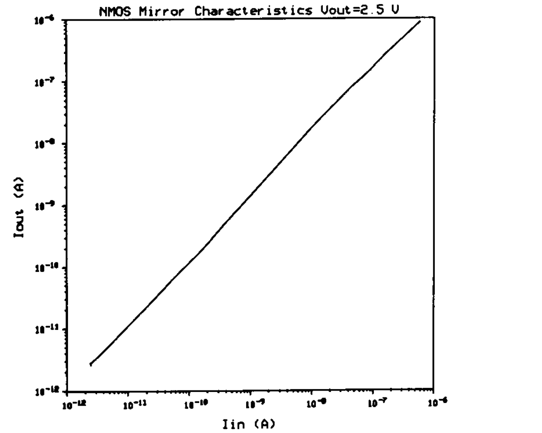


Fig. 4. Transfer characteristics of a minimum size current mirror circuit. Good mirroring is obtained for currents over five decades. As long as the devices operate in the subthreshold, mirroring is temperature independent.

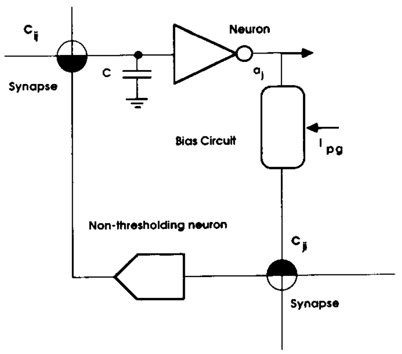


Fig. 5. A simple synthetic neural circuit. The synapses are programmable transconductances and the capacitance in the input of the neuron is that of the interconnect line.

3.2. *Current Mirrors*

A diode-connected transistor (drain and gate shorted) serves as a current-to-voltage converter, generating an output voltage that is applied to identical transistors to produce copies of the input current (see Fig. 3(a)). These transistors *mirror* the input current when they are operating in the saturation region. However, variations in substrate voltage, geometry, or doping can produce variations in the output current [16]. A current mirror is the simplest example of a CM circuit. It is our primary computational element. In addition to replicating currents, the mirroring operation is used to invert and to scale currents (see Fig. 3(b)). We have experimentally verified subthreshold mirror operation over several decades of current (see Fig. 4). N- and P-current mirrors can be cascaded because their input and output currents have compatible directions and the input conductance, g_m, is much larger than the output conductance, $g_{d\,\mathrm{sat}}$ (refer to (6) and (5)).

IV. Synthetic Neural Circuits

Fig. 5 shows a simple synthetic neural circuit. Two types of neurons are shown—a thresholding and a nonthresholding neuron. These neurons communicate with each other

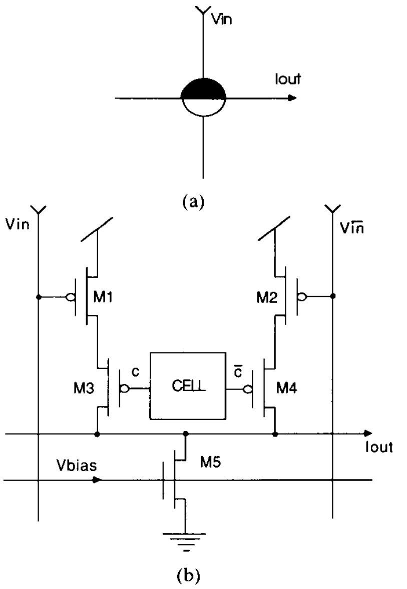


Fig. 6. (a) The symbol for a synapse and (b) its actual implementation. All transistors in the synapse are minimum size (3 μm × 6 μm).

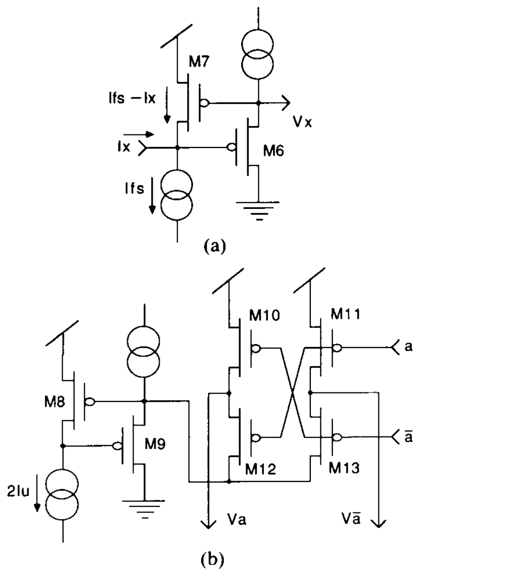


Fig. 7. Circuit diagrams for (a) the non-thresholding neuron and (b) the bias circuit. The output transistors are not minimum size.

through synapses as indicated. The neurons apply voltages to the synapses which, in turn, feed currents to the neurons. The half-filled disk symbol for the synapses was chosen to reflect this. Input voltages are applied to the dark half of the disk while the output current is obtained on the line separating the two hemispheres. The input line of a neuron may run through several synapses; the synaptic currents simply sum together. The bias circuit allows the current levels to be externally programmed. We now outline the operation of each of these elements and describe their circuit realizations.

4.1. Thresholding Neurons

Thresholding neurons a_i are simply MOS inverters. They receive bipolar current inputs from the synapses. These currents are integrated over time by the interconnect capacitance, thus the voltage on the capacitance represents activation. Neurons switch to the $+1$ state (or the -1 state) when this voltage exceeds (falls below) the inverter's threshold ($V_{\text{inv}} \approx V_{\text{dd}}/2$), and remain in the same state when the net input current is zero. Thresholding neurons drive the synapses through the bias circuit.

4.2. Synapses

The output current of a synapse is given by

$$I_{\text{out}} = cI_{\text{in}} \tag{7}$$

where $c = \pm 1$ is the state of the synapse. The input current I_{in} may have either direction. Thus the synapse performs a (four-quadrant) multiplication by a one-bit weight. The circuit used is shown in Fig. 6(b). Instead of supplying the input current I_{in} directly to the synapses it is encoded as a pair of voltages V_{in} and $V_{\overline{\text{in}}}$. These voltages are applied to the gates of transistors M_1 and M_2. V_{in} is set to obtain a drain current of $I_{\text{DC}} - I_{\text{in}}$ in M_1 while $V_{\overline{\text{in}}}$ biases M_2 to supply $I_{\text{DC}} + I_{\text{in}}$. I_{dc} is simply a dc shift introduced to guarantee that the currents in M_1 and M_2 are unidirectional. It is removed at the output by M_5 wh:ch is biased (using V_{bias}) to sink I_{dc}. This scheme allows I_{in} to be replicated in several synapses using the same lines.

The state c of the synapse is represented by a voltage at GND(-1) or at $V_{\text{dd}}(+1)$ in the memory cell. In the former case, M_3 is on and M_4 is off so that M_1 and M_3 together supply $I_{\text{DC}} - I_{\text{in}}$ to the output node. In the latter case, the reverse is true, hence M_2 and M_4 supply $I_{\text{DC}} + I_{\text{in}}$. Clearly, if M_5 subtracts I_{dc} the desired operation is obtained (7).

To compute inner-products bit-wise comparisons (multiplications) are required. The desired output from the synapses is

$$I_{\text{out}} = caI_u \tag{8}$$

where a and c are the states of the thresholding neuron and the synapse, respectively. This demands that the bias circuit set the voltage inputs such that $I_{\text{DC}} = I_u$ and $I_{\text{in}} = I_u$. The inner-product is obtained in units of $I_u \equiv 1$ by summing the output currents from all the synapses involved.

On the other hand, for the weighted sums required to compute activation, the desired output is

$$I_{\text{out}} = cI_x \tag{9}$$

where I_x is the input current to the nonthresholding neuron. This demands that $I_{\text{in}} = I_x$ and $I_{\text{DC}} = I_{\text{fs}}$, where I_{fs} is the full-scale current. The input voltages must be set accordingly by the nonthresholding neuron.

4.3. Bias Circuit

Given the state a of the thresholding neuron and an externally programmed current level I_{pg} the bias circuit, shown in Fig. 7(b), generates the required voltages for the

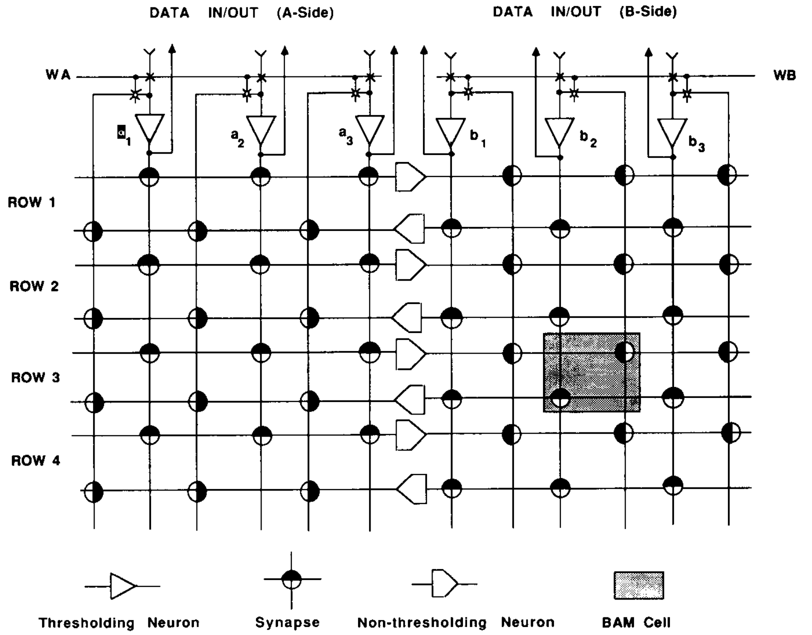


Fig. 8. Three-layer BAM chip architecture. The BAM cell is replicated to produce networks of any size.

synapses. Its outputs V_a and $V_{\bar{a}}$ drive the inputs V_{in} and $V_{\overline{\text{in}}}$, respectively. The circuit operates as follows: The current in M_8 is set to I_{pg} by feedback through M_9 which senses and corrects any current imbalance. The outputs are switched between V_{dd} and the voltage at the gate of M_8 using the multiplexer formed by $M_{10}-M_{13}$. If a is high $(+1$ state), V_a is tied to V_{dd} while $V_{\bar{a}}$ equals the voltage at the gate of M_8. If a is low $(-1$ state), the reverse is true. Transistors M_9-M_{13} are sized-up devices which have the necessary fan-out capability. By setting $I_{\text{pg}}=2I_u$ and V_{bias} to sink I_u through M_5 the desired synapse operation (see (8)) is obtained.

4.4. Nonthresholding Neuron

Nonthresholding neurons accept a bipolar input current I_x and generate the output voltages V_x and $V_{\bar{x}}$ which drive the synapses. The circuit used is shown in Fig. 7(a). This circuit is similar in operation to the bias circuit. It generates V_x which is applied to the input V_{in} of the synapse. V_x biases M_1 to source $I_{\text{fs}}-I_x$, mirroring the current in M_8. An identical circuit is fed $-I_x$ to obtain $V_{\bar{x}}$ which biases M_2 (in the synapse) to source $I_{\text{fs}}+I_x$. With V_{bias} set to sink I_{fs} in M_5 the desired output relationship (9) is obtained.

These functions have been implemented with a few devices using simple circuit configurations. This, plus the fact that all transistors in the synapses are minimum-size ($3\ \mu\text{m}\times 6\ \mu\text{m}$), makes their accuracy highly dependent on the fabrication process, i.e., variations of g_m, $g_{d\text{sat}}$, and I_0. The bias circuits and nonthresholding neurons use sized-up output devices with the appropriate fan-out capabilities. The rationale behind this approach is that by studying the short-comings of these simple circuits we can justify any additional complexity and thereby develop an efficient design methodology.

V. Implementation

5.1. Architecture

Our architecture is based on a regular array of BAM cells. Each BAM cell consists of two synapses and a one-bit memory cell. This pair of synapses provides two-way communication (bidirectionality) between neurons in the input/output layers and the middle layer. The bit stored in the memory cell determines the state of both synapses (symmetry). Fig. 8 shows a 3×3 BAM that stores up to four associations (one vector pair per row). This figure illustrates how neurons in the three layers communicate through the BAM cells. The input and output lines of the thresholding neurons at the top run vertically, while those of the nonthresholding neurons in the middle run horizontally. In general, communication in a BAM with n neurons in each input/output layer and $2s$ middle-layer neurons is supported by two $n\times s$ BAM cell arrays. Obviously, the number of neurons in the input/output layers need not be the same.

For every association programmed, a vector is stored in each BAM array at the same row. These vector pairs, (A^p, B^p), are stored in the BAM cells as follows:

Bit a_i^p (or b_j^p) of vector $A^p(B^p)$ is stored in the BAM cell at row p and column $a_i(b_j)$.

In the recall mode, the input vector is presented to one side, for example the A-side, and the WA signal is asserted. This initializes the state of the A-neurons (refer to Fig. 8). At the same time, the feedback is decoupled, allowing the A-neurons to launch the network toward the desired stable state. After *WA* is de-asserted, the network relaxes. To see that the operation is indeed as defined in (1) observe that the vectors in the BAM cells are compared in parallel with the input vector by the synapses. Each *output* synapse[2] does a bit-to-bit comparison, sourcing current onto (or sinking current from) a summing-line if there is a match (mismatch). (See (8). These currents sum to give the inner-products that are fed to the non-thresholding neurons, that is

$$u^p = \sum_{i=1}^{n} a_i^p a_i I_u.$$

The input synapses on the *B*-side now output (9):

$$I_{\text{out}} = b_j^p u^p.$$

Summing these currents and substituting the expression obtained for u^p, with $I_u = 1$, we obtain neuron j's activation as

$$v_j = \sum_{p \in \Omega} b_j^p u^p = \sum_{p \in \Omega} b_j^p \sum_{i=1}^{n} a_i^p a_i$$

which is simply (1). Activation is computed similarly for the *A*-neurons. When both *WA* and *WB* are de-asserted this process occurs simultaneously in both directions.

5.2. *Performance*

In this section we discuss the effects of dynamics on recall rates and describe a chip implemented using the architecture described. The fabrication and testing of these chips is also discussed.

To determine how fast the network relaxes, consider the large-signal response of a current mirror:

$$\frac{I_{\text{out}}}{I_0} = \frac{I_{\text{in}}}{(I_{\text{in}} - I_0)e^{-g_m t/C} + I_0}; \; g_m|_{I_{\text{in}}} = \frac{I_{\text{in}}}{\gamma U_T} \tag{10}$$

where C is the input-line capacitance. This yields an output current rise time of

$$t_r = 4.4C/g_m = 4.4\gamma C U_T/I_{\text{in}} = 4.4\gamma U_T/S \tag{11}$$

where $S = I_u/C_{\text{syn}}$ is the rate at which each synapse charges its local capacitance $C_{\text{syn}} \approx 100$ fF. Thus for $I_u = 0.5$ μA we find $S = 5$ V/μs and $t_r = 46$ ns.

The delay is obtained from (10) as

$$t_d = \gamma U_T \ln(I_{\text{in}}/I_0 - 1)/S \tag{12}$$

With $I_{\text{in}} = 10$ μA (full-scale current) and $I_0 = 1$ fA we find $t_d = 0.24$ μs. These results predict about 0.3 μs delay when the output synapses drive the nonthresholding neurons.

The bias circuits, the nonthresholding neurons and the input synapses drive purely capacitive loads; each global line has about 5 pF capacitance. The speed is limited primarily by the input synapses which must drive the inputs of the inverters (thresholding neurons) from V_{dd} or GND to V_{inv}. Assuming a current drive of 10 μA, these stages together have a delay of about 2.2 μs. Thus signals propagate from one input/output layer to the other and back in about 5 μs.

Observe that, for a given synaptic current level, t_r does not depend on the size of the network. t_d is much larger than t_r because g_m decreases with the input current. It can be reduced by decreasing the ratio I_{in}/I_0. Performance may be further improved by using a more sophisticated nonthresholding neuron whose time response depends only on its local parasitic capacitance and not that of the global interconnects. Such a "neuron" has been designed and will be used in another version of the chip.

In our prototype, both unit and full-scale currents are externally programmed. This option was added, at a small expense in area, to allow us to investigate the power/speed trade-off. In a future version, unit and full-scale currents could be generated using on chip bias generators.

The chips were fabricated by MOSIS [17] (production run M83I-IMOGENE) in 3 μm p-well CMOS technology. A microphotograph of the die is shown in Fig. 9. The die size is 2.3 mm × 3.4 mm with 4.8 mm^2 of useful area and 7200 transistors. Functional units were obtained on the first run. There are 32 thresholding input/output neurons (sixteen on either side), 14 non-thresholding neurons and seven 32-bit shift registers to store the vector pairs. A 16-bit input/output and control bus runs across the top of the chip. In the store mode, the bus is used to load data into the shift registers. The new data are stored in the top register while old data shift downward to the next row. In the recall mode the states of thresholding neurons on either side are initialized and read through the bus. Out of 20 dice received, 1 die was rejected during visual inspection and ten have been bonded and found to be functional. We have been able to store three nonorthogonal vectors, and successfully recall both the vectors and their complements from either side. With unit and full-scale currents of 0.5 and 9.5 μA, respectively, the network relaxes in less than 10 μs. The chip is able to perform error correction and recall on corrupted data. Complete test results will be duly reported.

VI. Conclusions

We have designed and fabricated a dense, repeatedly programmable, neural model for a heteroassociative memory. We obtain high density by using local storage at the expense of fault-tolerance. However, we can store two or three copies of each vector and still use less digital memory than a distributed matrix scheme. Higher order neural networks may be implemented by modifying the nonthresholding neurons.

CM circuits operating in the subthreshold region are used to achieve large fan-in and fan-out and low power dissipation. A scalable architecture results from employing coupling elements in a highly regular structure and avoiding the use of resistors. The speed of our network is limited by the ability of each synapse to charge/discharge its local parasitic load and not by the size of the network. By keeping voltage swings small we obtain fast operation. Using current inputs allows the interconnects to perform

[2] We refer to synapses on the outputs of the thresholding neurons in this fashion and those on their inputs as *input* synapses.

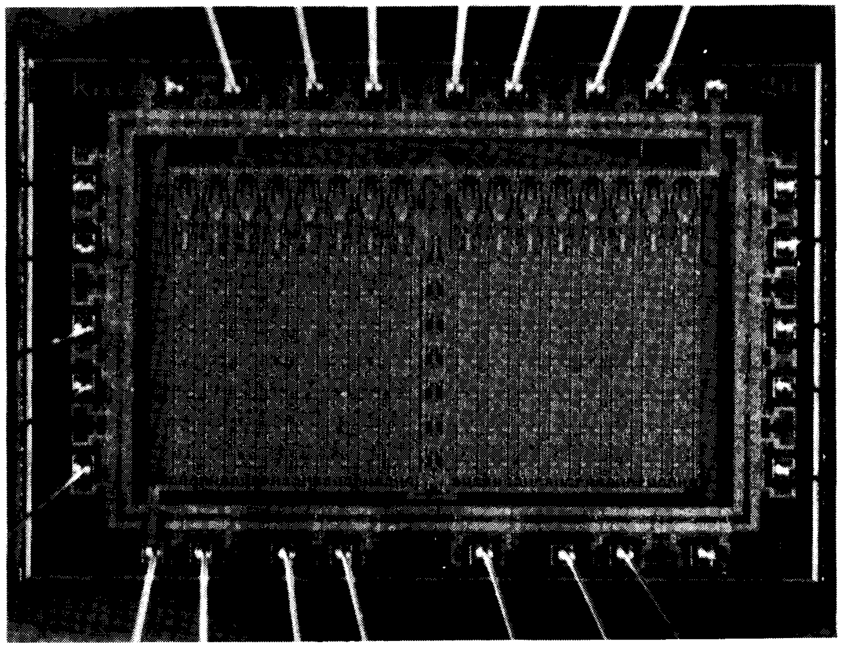


Fig. 9. Die microphotograph. The degree of regularity and density obtainable with this architecture is evident.

useful computation, and thus permits more efficient use of the silicon. It is evident from the die microphotograph that the BAM cell size is limited by the pitch of the second level metal lines. Therefore, to obtain higher functionality we must utilize the wiring even more. Such schemes have been developed and are included in the next generation of associative processors.

The system described in this paper has evolved around a simple principle: "*Communication is Computation*." Perhaps that is how biological information processing systems circumvent the bottlenecks of traditional computing.

Acknowledgment

The authors would like to thank Prof. C. R. Westgate for his support, Kim Strohbehn for helping out with the CAD tools, Aleksandra Pavasovich for providing experimental data on the behavior of MOS current mirrors, and Fernando Pineda for critically reviewing the theory. Special thanks are due to Prof. Carver Mead for making available a preprint of [9], used as the text for the analog VLSI class at Johns Hopkins, which inspired many of our ideas. This chip was a class project whose fabrication was funded by the National Science Foundation.

References

[1] J. P. Wade and C. G. Sodini, "A dynamic cross-coupled bit-line Content Addressable memory cell for high-density arrays," *IEEE J. Solid-State Circuits*, vol. SC-22, pp. 119–121, 1987.

[2] S. R. Jones, I. P. Jalowiecki, S. J. Hedge, and R. M. Lea, "A 9-kbit associative memory for high-speed parallel processing applications," *IEEE J. Solid-State Circuits*, vol. 23, pp. 543–548, 1988.

[3] N. Carriero and D. Gelernter, "Applications experience with Linda," in *Proc. ACM/SIGPLAN Symp. on Parallel Programming*, pp. 173–187, July 1988.

[4] T. Kohonen, *Self-Organization and Associative Memory*. New York: Springer Verlag, 1988, (2nd edition).

[5] ____, *Associative Memory: A System Theoretic Approach*. New York: Springer Verlag, 1977.

[6] J. J. Hopfield, "Neural networks and physical systems with emergent computational abilities," in *Proc. Nat. Acad. Sci. USA*, vol. 79, pp. 2554–2558, 1982.

[7] T. Kohonen, *Content-Addressable Memories*. New York: Springer Verlag, 1980.

[8] M. A. Sivilotti, M. R. Emerling, and C. A. Mead, "A novel associative memory implemented using collective computation," in *1985 Chapel Hill Conf. Very Large Scale Integration*, Henry Fuchs, Ed., Comp. Sci. Press, 1985.

[9] C. A. Mead, *Analog VLSI and Neural Systems*. Reading, MA: Addison-Wesley, (in press).

[10] Mary Ann C. Maher, S. P. DeWeerth, M. Mahawold, and C. A. Mead, "A methodology for implementing neural architectures," pp. 000–000, this issue.

[11] B. Kosko, "Bidirectional associative memories," *IEEE Trans. Syst. Man. Cybern.*, vol. 18, pp. 49–60, Jan./Feb. 1988.

[12] E. B. Baum, J. Moody, and F. Wilczek, "Internal representation for associative memory," *Biol. Cybern.*, vol. 59, pp. 217–228, 1988.

[13] E. A. Vittoz and J. Fellrath, "CMOS analog integrated circuits based on weak inversion operation," *IEEE J. Solid-State Circuits*, vol. SC-12, pp. 224–231, June 1977.

[14] A. G. Andreou, K. A. Boahen, and P. O. Pouliquen, "An automated data acquisition system for testing MOS devices and analog circuits in the subthreshold region," *IEEE Trans. Inst. Meas.*,

[15] Mary Ann C. Maher and C. A. Mead, "A physical charge-control model for MOS Transistors," in *Advanced research in VLSI: Proc. 1987 Stanford Conf.* Paul Losleben, Ed., the MIT press, pp. 211–229, 1987.

[16] A. Pavasovich, A. G. Andreou, and C. R. Westgate, "An investigation of minimum-size, nano-power, MOS current mirrors for analog VLSI systems," JHU Elect. and Comp. Eng. Tech. Rep. JHU/ECE 88-10.

[17] D. Cohen and G. Lewicki, "MOSIS—The ARPA silicon broker," in *Proc. Second Caltech Conf. VLSI*, pp. 29–44, Pasadena, CA, 1981.

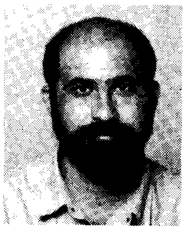
Kwabena A. Boahen is working toward the B.S./M.S.E. degree in electrical engineering at the Johns Hopkins University, Baltimore, MD.

His current research interest are Analog VLSI design and testing, with applications in synthetic neural and sensory systems.

Mr. Boahen is a member of Tau Beta Pi.

Philippe O. Pouliquen is working toward the B.S. degree in biomedical engineering at the Johns Hopkins University, Baltimore, MD.

His current research interests are instrumentation, and microcomputer and VLSI applications in neurology.



Andreas G. Andreou (S'81–M'86) received the M.S. and Ph.D. degrees in electrical engineering from The Johns Hopkins University, Baltimore, MD, in 1983 and 1986, respectively.

Upon graduation he was a Post-Doctoral Fellow and subsequently Associate Research Scientist in the Department of Electrical and Computer Engineering and the Applied Physics Laboratory of The Johns Hopkins University. His current research interests are in the areas of device physics and characterization, solid-state sensors, and analog VLSI.

Dr. Andreou is a member of Tau Beta Pi.

Robert E. Jenkins (M'86) received the B.S. degree in engineering and the M.S. degree in physics from the University of Maryland, College Park.

At present he is a lecturer in the School of Engineering and a staff engineer at the Applied Physics Laboratory, Johns Hopkins University, MD, leading the Space Department IR&D program. He is also a member of the program committee for the part-time M.A. program in electrical engineering.