An Analog Neural Computer with Modular Architecture for Real-Time Dynamic Computations

Jan Van der Spiegel, *Senior Member, IEEE,* Paul Mueller, David Blackman, *Member, IEEE,* Peter Chance, Christopher Donham, *Member, IEEE,* Ralph Etienne-Cummings, and Peter Kinget, *Student Member, IEEE*

Abstract—**The paper describes a multichip analog parallel neural network whose architecture, neuron characteristics, synaptic connections, and time constants are modifiable. The system has several important features, such as time constants for time-domain computations, interchangeable chips allowing a modifiable gross architecture, and expandability to any arbitrary size. Such an approach allows the exploration of different network architectures for a wide range of applications, in particular dynamic real-world computations. Four different modules (neuron, synapse, time constant, and switch units) have been designed and fabricated in a 2-μm CMOS technology. About 100 of these modules have been assembled in a fully functional prototype neural computer. An integrated software package for setting the network configuration and characteristics, and monitoring the neuron outputs has been developed as well. The performance of the individual modules as well as the overall system response for several applications have been tested successfully. Results of a network for real-time decomposition of acoustical patterns will be discussed.**

I. Introduction

RECENT years have seen a renewed interest in neural networks for several reasons: a better understanding of the functions of the brain [1], improved mathematical models [2], development of new algorithms and net topologies, and the understanding of emergent collective properties of neural networks [3]–[5]. While much of the work has been theoretical, technological advances are beginning to make hardware implementations of such systems possible with the goal to build machines capable of solving real-world problems. These tasks require computational power that is beyond the capabilities of current von Neumann-based digital computers. Biological systems solve these problems through parallel computations in a highly interconnected structure consisting of many nonlinear processing elements.

Several approaches have been taken to the implementation of neural networks. Such efforts date back to the early 1960's when electrical models of simple neurons and also more complex neural systems were first built [6], [7]. Only recently have more powerful and compact systems evolved. Technologies that are being used today include digital systems [8]–[10], either in serial or parallel form, analog electronic networks [11]–[21], and optical systems [22]–[23]. Digital approaches range anywhere from simulation on von Neumann machines to special-purpose digital neural networks. The advantages of digital approaches are programming flexibility, computational accuracy, noise immunity, as well as the fact that digital technology is well established. However, digital methods have some distinct drawbacks which are inherent in the essentially sequential nature of the computations, and which remain even in a highly parallel architecture. For example, though high-speed parallel computers can calculate the state of each neuron in parallel, neurons still evolve sequentially from state to state. The digital computer calculations must converge for each time slice. Since the neural network elements are nonlinear elements of input and time, convergence can be a problem for networks configured with feedback and lateral inhibition. Another serious bottleneck in the performance of digital computer networks is the bandwidth between processors and memory. In order to calculate the input for a given neuron, a processor must have access to all other neuron outputs. Hence, each processor must be able to access a large shared memory where the previous neuron states are stored. An extremely large bandwidth is required so that the memory can communicate with the processors.

In contrast, analog systems can sum and scale arbitrary numbers of inputs truly simultaneously. The limited accuracy of analog components is not a serious problem because neural networks are forgiving to component errors for the following reasons: 1) the output of the network is the aggregate response of many neurons acting collectively, and thus the result is less dependent on minor component variations; and 2) learning can be used to tune the

Manuscript received May 27, 1991; revised August 27, 1991. This work was supported by the Office of Naval Research under Grant N0014-89-J-12499, the National Science Foundation under Grant EET 16685, and Corticon, Inc. C. Donham and P. Kinget were supported by the National Science Foundation through Research Experience Undergraduates Grants ENG88-05282 and ENG89-00442.

J. Van der Spiegel, P. Mueller, D. Blackman, C. Donham, and R. Etienne-Cummings are with the Departments of Electrical Engineering and Biophysics, University of Pennsylvania, Philadelphia, PA 19104-6390.

P. Chance is with Corticon, Inc., Philadelphia, PA 19104.

P. Kinget is with the Laboratory Electronika, Systemen, Automatisatie, Technologie (ESAT), Departement Elektrotechniek, Katholieke Universiteit Leuven, B-3001 Heverlee, Belgium.

IEEE Log Number 9104598

network response in spite of specific component error. Another advantage of analog systems is that time can be easily represented as a continuous variable through a time to potential or current transformation, thereby avoiding the clocks and counters required in digital systems to keep track of time. As a result, analog neural networks can provide not only a mechanism for summation and scaling of incoming signals, but for temporal integration as well, in a fashion similar to biological networks. Biology has realized the advantages of analog methods and has achieved the impressive computational rates that approach the equivalent of 10^{18}–10^{20} floating-point operations per second[1] in spite of its slow hardware.

This paper describes a large-scale neural network that is loosely modeled after the biological system. It is fully parallel and operates in analog mode. The architecture, the neuron characteristics, synapses, and synaptic time constants are modifiable. The system has several important attributes: 1) in addition to synaptic weights that are adjustable over a large dynamic range, the system contains modifiable synaptic time constants which are required for time-domain computations; as a result the network does *spatio-temporal* processing, which is a considerably more complex task than parallel processing done at discrete time steps; 2) the system is constructed from interchangeable modules with two-dimensional symmetrical pinout allowing a modifiable gross architecture in which the numerical ratios and arrangement of neurons, synapse, and connection modules can be easily selected; 3) except for global control lines, the modules have only connections to their nearest neighbors, resulting in a simplified packaging and board design; and 4) the network is expandable to arbitrary size and can therefore be adapted to the complexity of the task.

The network can be used in both learning and computation mode. However, learning algorithms are implemented through a host computer by sampling the network state (as described in Section IV), and reprogramming the network as required. Thus, the full potential of the machine is obtained, after learning, in situations involving neural computation of dynamic systems, such as acoustic pattern recognition, sonar signal classification, motion detection, etc. A prototype system consisting of 99 custom chips has been designed, assembled, and tested. The overall system is fully operational and has been successfully used for several dynamic computations. In addition, supporting software for programming the network and monitoring the state of the net has been developed. Results and performance of the individual modules and the overall system are given below.

[1]There are about 10^{11} neurons in the human brain which are richly interconnected with feedback and lateral inhibitions. Each neuron has on the average 1000 synapses resulting in a total of 10^{14} connections. Each connection has a time constant of approximately 1 ms which corresponds to the summing of charge on the postsynaptic membrane capacitance. To simulate this set of simultaneous nonlinear differential equations describing such a system, a time step as small as 0.1 ms would have to be used. Thus, 10^{18} postsynaptic potentials would be updated per second. If each potential update took 1 to 100 floating-point operations, the effective processing rate would be 10^{18}–10^{20} FLOPS, assuming all neurons are active.

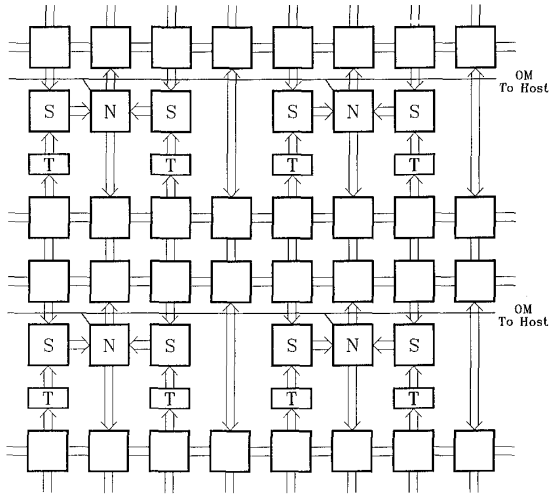

Fig. 1. Block diagram of the modular gross architecture of the neural network. The system consists of programmable neuron modules (N), synapse modules (S), time constants (T), and analog crosspoint arrays (blank boxes). The modules are directly interconnected through parallel analog lines. The state of each neuron is sampled and multiplexed on a single line (OM) that is connected to a digital host.

II. System Architecture

As shown in Fig. 1, the machine architecture is a multichip system containing neurons, synapses, modifiable time constants, and analog crosspoint switches. The approach is to a certain extent inspired by biological systems, in particular the cerebral cortex, in which there are separate neurons, axons, and synapses with a limited number of inputs for each neuron. In contrast to the biological system, the system described here is fully modifiable including the neuron, synapse, and synaptic time constants, as well as the network topology. Such a design will allow the exploration of different network architectures for a wide range of applications, in particular dynamic real-world problems.

The modules of the neuron arrays are placed on a regular grid and connected directly to neighboring synaptic and analog switch arrays. During operation the analog signal flow is as follows: the output signals exit from the north and south of the neuron chip; then the signals are routed via the analog switches until they are steered north or south into synapse modules, through a time constant module if required; the synapses provide the input to the neighboring neurons along the east or west direction. The switch modules allow the neuron outputs to be routed to any other neuron. Hence, fully interconnected multilayered networks can be easily implemented. However, the number of neurons that can be fully interconnected between layers is limited to the number of synaptic inputs per neuron. Also, a wide range of sparsely interconnected networks can be mapped into the system. However, in

general, the machine architecture presents more opportunities for short-range connections. Thus, the architecture favors local computations and a hierarchical organization of data flow, which also seems to be true for the biological system.

The ratio between the number of neurons, routing channels, and synapses on each module is chosen to optimize the usage of available components for typical applications. Earlier experience with a discrete system based on a similar architecture for the neural computation of acoustical patterns [24] has served as a guide in the choice of these module sizes. However, the modules are designed so that they can be repeated where required (e.g., placing two synapse modules on each side of the neurons to double the fan-in, or using additional rows of switch modules to increase the routing capabilities).

The system operates fully analog and in parallel. However, the switch positions, neuron characteristics, synaptic weights, and time constants are set by a digital host and are stored in a local memory on the individual modules. In addition, all or a selected number of neuron outputs can be sampled continuously or during a specified time segment and multiplexed on a single line (called the output monitor line or OM) which is connected to an A/D converter and read into the digital host (Fig. 1). This operation, which is independent of the analog neural computations, allows the host computer to display the outputs as a function of time, or to use them for implementing learning algorithms. The interaction between the neural network and the host computer is described in Section IV.

III. Description and Performance of the Modules

Four different modules have been designed and fabricated in a 2-μm n-well CMOS technology through the MOSIS service.[2] The function, major design issues, and results of each chip will be described. These chips are used in a prototype system of 72 neurons whose overall performance is discussed in Section IV. For this prototype, synapse arrays of 16 $\times$ 8 synapses, switch arrays of 16 $\times$ 16 switches, and neuron chips of eight neurons were fabricated. The number of synapses and switches is chosen so that the overall size of each die is not excessive, yet the chips are complex enough to allow evaluation of the performance of the whole system. The choice of eight neurons per module gives an optimum ratio between neurons and routing channels, as mentioned in Section II. By using larger dies and more advanced packaging methods, the number of components per module can be scaled up.

A. Neuron Module

This module consists of individual neurons, an analog multiplexer, and digital control logic for addressing the chip and driving the multiplexer. For the prototype system we choose eight neurons per module. Fig. 2 shows a schematic block diagram of the neuron chip.

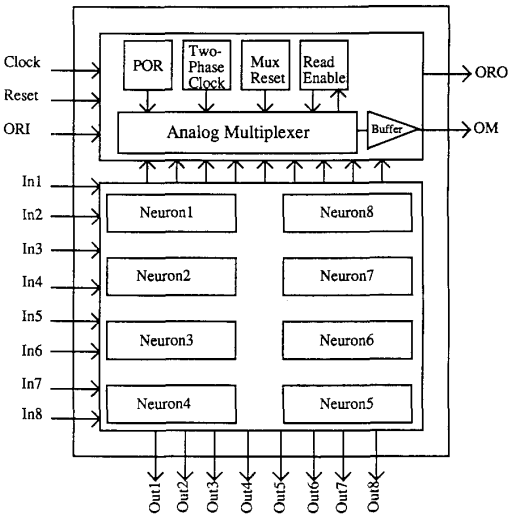

Fig. 2. Block diagram of a neuron module. Each unit consists of modifiable neurons, an analog multiplexer, and control logic for addressing the chip.

The neuron is implemented as a piecewise-linear device having a transfer function with a variable magnitude of the step at threshold, and variable threshold current. The input signal is a current and the output is a voltage. This representation is chosen because currents can be easily summed and scaled while voltage outputs have a large fan-out. Hence, the input op amp is used in a transimpedance configuration, with a rectifier, and the output op amp is a unity-gain inverting amplifier, as shown in Fig. 3. The transimpedance gain has to be matched to the synapse characteristic that provides the input current. The gain is 0.1 V/μA using nominal resistance values of 100 kΩ. The diodes are implemented as diode-connected MOS transistors. The comparator, which triggers the step when the input current exceeds the threshold value, is a PMOS-input differential pair. The step is provided by a current mirror, which is biased off chip by the synapse module.

The network is designed for real-world applications where the input signals have frequency components limited to tens of kilohertz. This is taken into account in the design of the neurons by intentionally slowing them down. This has the additional advantage that crosstalk will be minimized in neighboring lines of the network, particularly in the switch chip. Special attention is paid to maintaining the stability of the op amps under all conditions. The stability is influenced by the input and output capacitances, which can be as large as 40 and 200 pF, respectively, the output impedance of the synapses, which is highly dependent on the selected weight, and the diode resistance in the feedback loop, which is also current dependent. Furthermore, the neuron must be able to provide an output voltage between 0 and 4 V. Miller op amps are used and are designed to have a minimum gain and unity-gain bandwidth of 25K and 150 kHz, respectively. The

[2]MOSIS Service is a fast prototyping service offering fast turnaround standard cell and full-custom VLSI circuits at low cost. MOSIS® is a registered trademark of the University of Southern California.

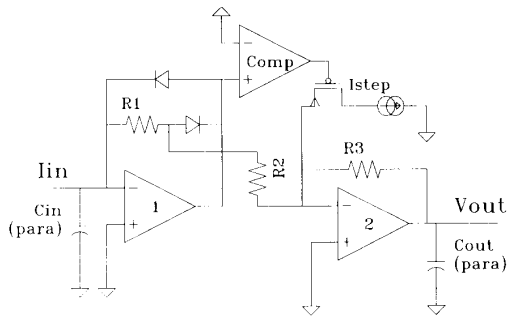

Fig. 3. Schematic of a neuron employed in the network. Each neuron has minimum output voltage at threshold, set by I_{step}.

comparator, on the other hand, requires a gain of 500 and unity-gain bandwidth of 750 kHz.

The analog multiplexer is designed with support circuitry to enable random sampling of a chip, and to generate protocol signals to indicate the completion of a sampling sequence. The multiplexer consists of an 8-b shift register, which activates transmission gates and reads the neuron outputs. These outputs are then presented to a buffer, to be read off chip via the OM line. The shift register is externally reset before every sampling sequence. The ORI line signals the start of a sample sequence while the ORO signals its completion. During standby, the buffer is disconnected from the multiplexer by a read-enable switch. The buffer is also designed as a Miller op amp, and is required to be stable for a large capacitive load and to be fast. However, a compromise was reached between these two requirements, and the simulated buffer is found to be slew-rate limited, having a slew rate of 0.5 $V/\mu s$. This is sufficiently fast for the applications of this network.

The measured neuron characteristics are found to be within the specifications. Each neuron has an area of 228 $\times$ 566 μm^2 and consumes 12.2 mW when operating between a -5- and $+5$-V power supply. The value of the resistors is 30% larger than the nominal design value. However, the standard variation of the resistors over a module and over different chips is 2%. The larger transimpedance gain can be easily compensated for by the synapses and does not pose a problem. The measured transfer characteristics of the neurons is shown in Fig. 4. Fig. 4(a) gives the output voltage versus input current from -60 to $+60$ μA for different minimum output voltages at threshold, ranging from 0 to about 5 V. Fig. 4(b) gives the neuron outputs for threshold currents (I_{step}) of -30 μA, 0, and $+30$ μA and a minimum output at threshold of about 1.7 V.

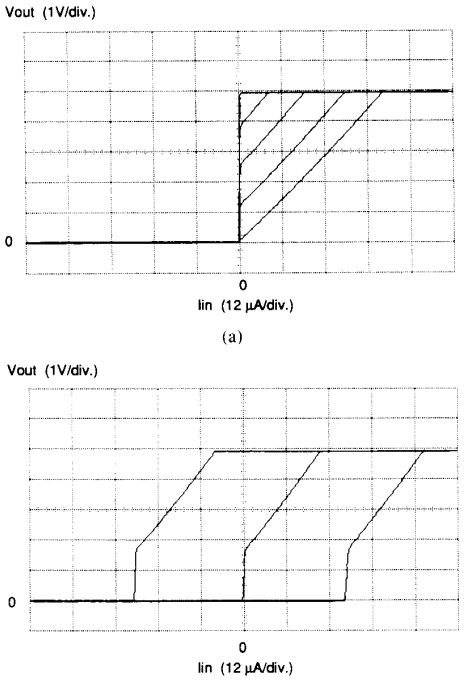

Fig. 4. Measured neuron output voltage versus input current. (a) Transfer characteristics for different minimum outputs at threshold ranging from 0 to 5 V. (b) Characteristics for three different threshold currents (-30, 0, and $+30$ μA) with a 1.7-V step. Vertical: 1 V/div., horizontal: 12 μA/div.

B. Synapse Module

The synapse module is made up of a variety of components (see Fig. 5): voltage-to-current converters, current splitters, current recombination units, shift registers, digital control logic, analog control logic, and offset control. The voltage-to-current (V-to-I) converter receives an input voltage from a neuron and produces an output of 10 μA/V. The current splitter multiplies the output of the V-to-I converter by several gain factors to produce the range of available output currents. The current recombination unit adds together an arbitrary combination of the splitter outputs, and produces the final output current that goes to the neuron. The shift register controls the current splitter based on a user programmed weight. The digital control logic is used in loading the shift register. Finally, analog control logic is needed to autobias both the V-to-I converter and the offset control.

The number of units required in each synapse chip is a large factor in determining how to implement the circuits outlined above. With neural networks expected to have 32 inputs and 16 outputs per module, the circuit for each synapse has to be designed to be relatively small. Another important factor is the signal swing. The neural network operates with signal voltages between 0 and 4 V, hence all transistors have to remain correctly biased for large input swings. Also, a large dynamic range of four orders of magnitude is required for the synaptic weights, which are distributed equally on a logarithmic scale. Lastly, as shown in Fig. 5, the V-to-I converter and current splitter are common to all synapses connected to the same input line. As a result, wherever possible, circuitry has been moved from the current recombination unit into the V-to-I converter and current splitter in order to minimize the area.

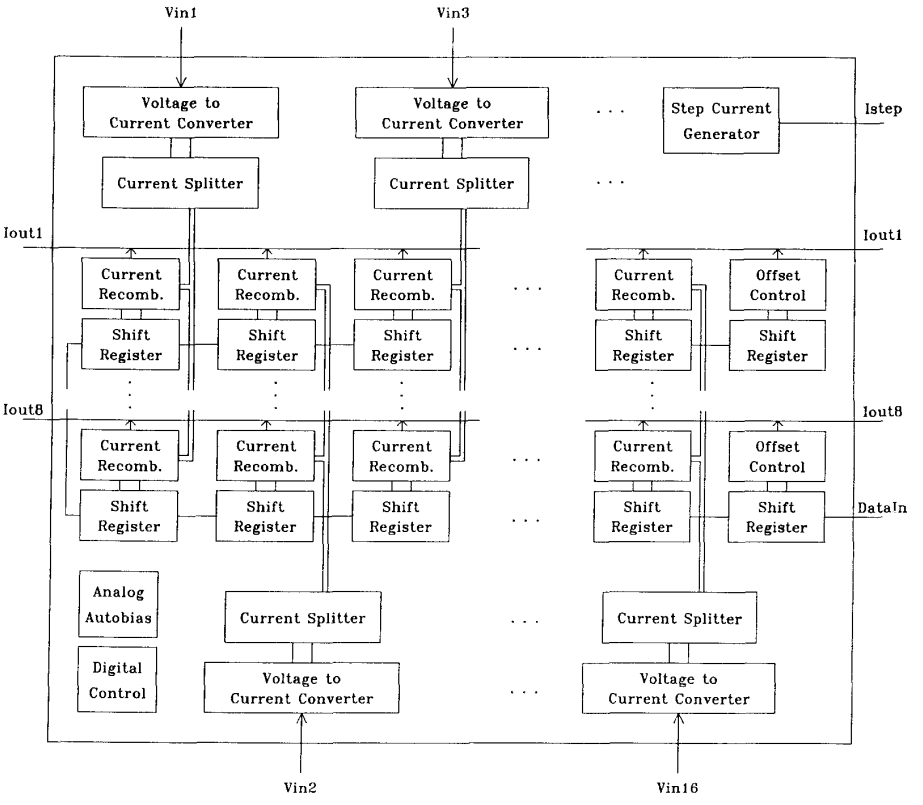

Fig. 5. Block diagram of the synapse module; the inputs are common to all synapses in the same column and the output lines sum the currents of all synapses in the same row.

The voltage-to-current converter is designed based on a six-transistor analog, four-quadrant multiplier developed by Bult and Wallinga [25]. This circuit provides a differential current that is dependent on the bias and the input voltages. The original design has been augmented to provide a single-ended output, to increase the output impedance and the current mirroring accuracy, resulting in a final design of 162 μm $\times$ 239 μm.

Two major methods are considered for implementing the current splitter and current recombination unit. In the first method, currents are divided into logarithmic units (i.e., 10, 1, 0.1, 0.01), and are passed to the current recombination circuit for further scaling (i.e., 1, 1/2, 1/4) to create the final range of synapse outputs. Such a design results in a minimal number of analog bus lines common to each synaptic input. The second method concentrates as much circuitry as possible in the current splitter, in lieu of circuitry in the current recombination circuit. All individual synaptic weights are generated in the current splitter (i.e., 10, 5, 2.5, 1, 1/2, 1/4, 1/10, 1/20, 1/40, 1/100, 1/200, and 1/400). Twelve analog bus lines are required for each synapse input. The second method is found to be significantly more compact for two reasons. First, as mentioned previously, since there are many more recombination units than splitters, a reduction in the size of the recombination unit at the sacrifice of space in the splitter is desirable. Second, the bus lines are routed over other circuitry in the synapse such that the additional bus lines do not require additional space in the recombination unit. There is another reason for using the second method: since scaling for a series of synapses is done in a single circuit, variation from synapse to synapse is significantly reduced, transistors can be sized more appropriately for the currents being scaled, and more accurate gain ratios can be made.

As shown in Fig. 6(a), the splitter is made of a series of ratioed current mirrors which results in effective current gains of 10, 5, 2.5, 1, 1/2, 1/4, 1/10, 1/20, 1/40, 1/100, 1/200, and 1/400. Each scaled current is connected to a diode-connected transistor, thus allowing a voltage to be passed to the recombination unit. Full-scale currents in these transistors range from 500 μA for a gain of 10 to 125 nA for a gain of 1/400. Size constraints limited the dimensions of some of the diode-connected transistors. For the weights 10, 5, 2.5, 1, 1/2, 1/4, 1/10, and 1/20, transistors with large widths and lengths could be used to improve matching between the splitter and the current recombination unit. However, transistors used for the smaller weights (1/40, 1/100, 1/200, and 1/400) could not be sized appropriately, hence these transistors will operate in weak inversion for small current flows. Therefore, the smaller weights are more suscepti-

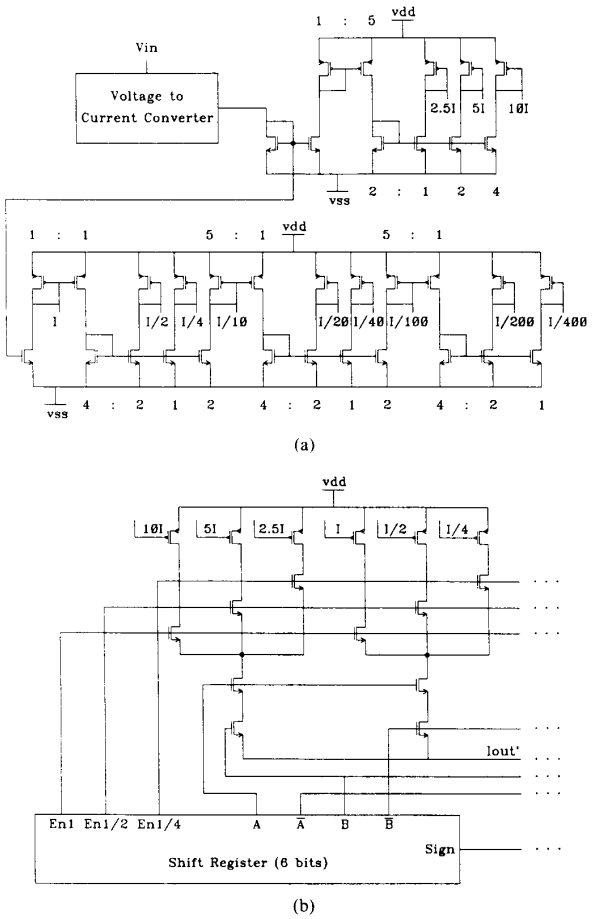

Fig. 6. Simplified circuit schematic of the synapse. (a) The top represents the V-to-I converter and the current splitter. (b) The bottom part is the recombination unit and shift register. Only the bottom part needs to be repeated for each synapse.

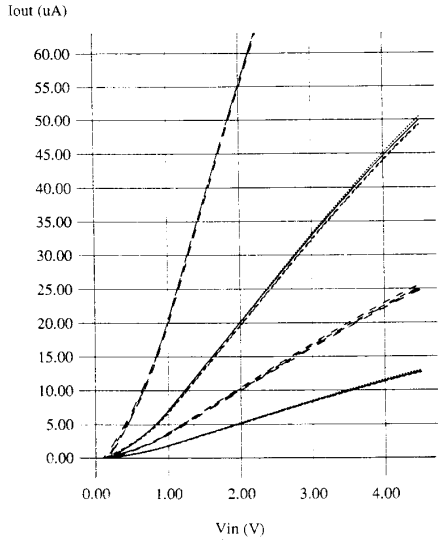

Fig. 7. Output current versus input voltage of a synapse, measured over four chips of the same run for weights of 2.5, 1, 0.5, and 0.25.

ble to noise and exhibit larger gain errors. The final size of a single layout containing both a V-to-I and a current splitter is 332 μm $\times$ 408 μm.

The current recombination unit is a relatively simple circuit as can be seen in Fig. 6(b). The voltage produced in the splitter on each line of the analog bus is converted back into a current via a PMOS transistor in saturation. NMOS transistors in linear mode are used to switch currents on or off to select the required weight.

A standard dynamic shift register is used to store the digital data associated with each synapse. Each bit has been modified slightly so that if the on-chip clock is locked in a certain phase, and an auxiliary control line is enabled, the shift cells become static. The shift register is designed to operate at 2.5 MHz. A maximum of 1 ms is allowed between operation in dynamic mode and static mode.

Each of the units described above has been fabricated and tested individually. Outlined below are some of the test results for the synapse module. Fig. 7 shows the output of a synapse with weights of 2.5, 1, 0.5, and 0.25 measured on four different chips. The V-to-I output ranges from approximately 1 nA for 0-V input to 45 μA for 4.0-V input. The curved shape at the lower limit of the characteristic and slight bowing in the center are caused by the V-to-I converter, as was verified in a test synapse where the output of the V-to-I converter was measured directly. Currently, a slightly curved synaptic response is considered beneficial for back-propagation learning algorithms, though future system testing will indicate whether the nonlinearity in the transfer characteristic will have to be modified.

In order to study the weight variations between synapses within a chip and between different chips, a total of about 1400 synapses were tested. A computer was used to scan for all weights for each synapse. The synaptic weights were determined by measuring the current output for two input voltages, and calculating the slope of the transfer characteristic. The results of these measurements for 128 synapses on a single chip, as well as the results of 1400 synapses over 11 chips of the same run are tabulated in Table I. As expected the error in matching is smallest for the range between 0.01 and 1 for which relatively large transistors could be used. The standard deviation for the inhibitory weights are somewhat larger than for the excitatory ones, which is due to the sign inversion mirror used to create the inhibitory weights. By comparing the results within one chip to those between chips one can conclude that the chip-to-chip variations are considerably larger, as expected.

The measured weights as a function of the synaptic code are shown in Fig. 8. These weights are normalized by the measured weight for a theoretical gain of 1. As can be seen, the normalized gains are very close to the theoretical ones. As mentioned previously, the neural network is sensitive to the ratio of the weights, not the absolute value.

TABLE I
MEAN VALUE AND STANDARD DEVIATION OF MEASURED SYNAPTIC WEIGHT VALUES OVER ONE CHIP (128 SYNAPSES) AND OVER A GROUP OF 11 CHIPS BELONGING TO THE SAME RUN

	Per Chip		Group of Chips	
Nominal Weight	Mean	St. Dev.	Mean	St. Dev.
10	10.32	0.12	10.43	0.44
5	5.29	0.066	5.34	0.78
2.5	2.62	0.026	2.62	0.12
1	1.000	0.010	1.000	0.22
0.5	0.492	0.005	0.493	0.096
0.25	0.251	0.003	0.252	0.024
0.1	0.101	0.001	0.100	0.021
0.05	0.0529	0.0006	0.052	0.011
0.025	0.0266	0.0004	0.026	0.0024
0.01	0.0111	0.0002	0.011	0.0024
0.005	0.0057	0.0002	0.006	0.0007
0.0025	0.0030	0.0002	0.003	0.0006
−0.0025	−0.0033	0.0003	−0.003	0.0010
−0.005	−0.0062	0.0006	−0.006	0.0019
−0.01	−0.012	0.001	−0.012	0.0035
−0.025	−0.029	0.002	−0.029	0.0076
−0.05	−0.057	0.004	−0.057	0.017
−0.1	−0.109	0.007	−0.108	0.024
−0.25	−0.270	0.012	−0.269	0.051
−0.5	−0.526	0.018	−0.52	0.134
−1	−1.062	0.026	−1.06	0.168
−2.5	−2.78	0.045	−2.78	0.346
−5	−5.65	0.081	−5.67	0.700
−10	−10.87	0.134	−9.59	0.441

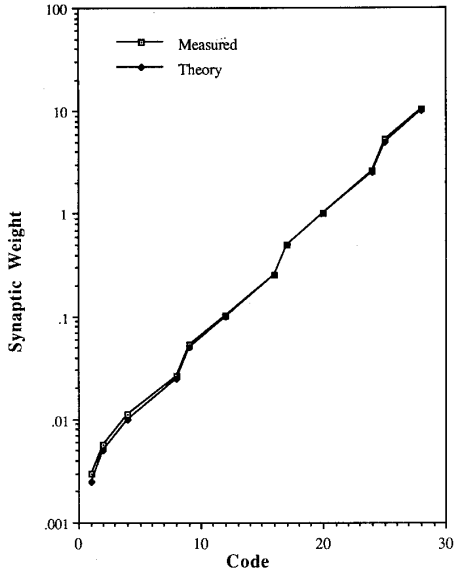

Fig. 8. Measured and theoretical weight factors of the synapse versus the digital code programmed in the synapse memory. The measured values are normalized by the measured weight value for a gain of 1.

Since the weight ratios are very close to the expected value, the synapse performance is more than adequate for proper operation of a neural network.

C. Time Constants

Modifiable time constants ranging between 5 and 1000 ms are required for time-domain computations. These time constants are implemented fully monolithically on a relatively small area. The values are set by a 4-b word, stored in a local memory. The time-constant value varies logarithmically with the digital code.

There are different ways to obtain these large time constants: a passive resistance and capacitor, a transconductance amplifier used as an active resistance, and switched-capacitor circuits. The first method would result in excessive chip area. The potential problems with the other methods are the large offsets and noise problems. The approach used here follows the second method and is based on a time-constant multiplication technique, by using a load-compensated operational transconductance amplifier (OTA) as the resistance. In the design of the circuit a very small current ratio between the input stage and output branch of the OTA is chosen [26]. This accomplishes two goals: first, it provides a large transconductance reduction; and second, it limits the small current to the output stage only and thus reduces the leakage current and the offset voltage. In addition, the input stage is designed to handle signals up to 4 V. Hence, the overall area of the amplifier and capacitance is minimized. The area of a 1000-ms time constant, including switching for selecting different time constants, memory for storing the digital code, the bias circuit, and an output buffer, is 1075 μm $\times$ 540 μm. Offset voltages in the range of 5 to 20 mV were obtained for time-constant values up to 60 ms and increased to about 200 mV for the larger time constants. Fig. 9 shows typical values of the measured time constants as a function of the digital code. The standard deviation of the time constants measured on 16 circuits over four chips lies between 2.5 and 7.5% depending on the time-constant value.

D. Crosspoint Switch Chip

The function of the switch chip is to route the analog signals between neuron, synapse, and time-constant modules. Each switch is set by 1 b, stored locally. When a switch is activated it connects a horizontal line to a vertical line. The interconnection architecture is thus set by the switch positions. A module consists of an array of such crosspoint switches. In addition, there are switches at the end of the horizontal and vertical lines. These switches interrupt the connection to the next switch, allowing the interconnection buses to be partitioned in several sections to optimize the available routing space. These switches can also ground unused lines in order to prevent floating lines. A 2-b memory is used to set each of these last switches.

The block diagram of a switch module consists of the switch fabric and of digital logic circuitry used to control loading of the memory. Each switch is realized as a CMOS transmission gate, clocked between $\pm$5 V. In making the floor plan and corresponding layout, special attention was paid to prevent latch-up in this module. Designing the input pads properly is critical in reducing the chance for latch-up in the switch chip.

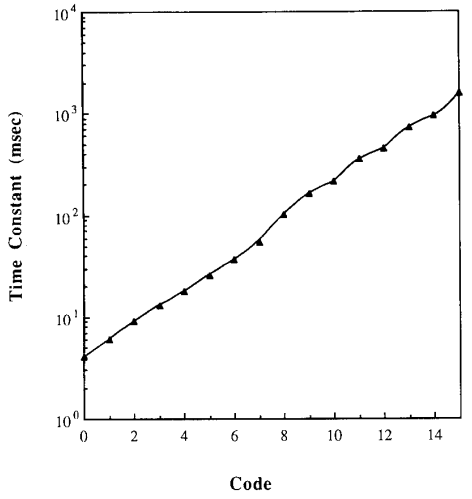

Fig. 9. Average time-constant value measured on 16 circuits. The standard deviation lies between 2.5 and 7.5%.

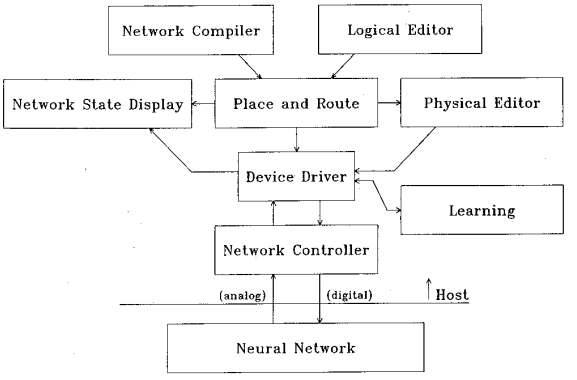

Fig. 10. Overview of the supporting software system, residing on a digital host, used to control the network configuration and to monitor the operation. The neural network can be programmed through a logical or physical editor or a network compiler. Learning algorithms are implemented on the host computer as well.

The switch modules have been found to be very reliable and no latch-up was found after taking the precaution, mentioned above. The on-resistance varies between 2 and 3 kΩ depending on the level of the input signal, which lies between 0 and 5 V. Off-resistance was in the teraohms range.

IV. Overall System Operation

The neural network is programmed from one of two perspectives: the physical view or the logical view (see Fig. 10). The physical editor displays a map of all the chips in the neural network. The user then selects a chip to edit, and makes the desired change to a specific synapse, switch, or time constant on the indicated chip. When changes are complete, the physical editor uses a device driver to send the chip settings to the network controller, which in turn drives the digital control wires appropriately to program the network.

The logical view displays an abstract representation of a neural network in a graph form using symbols for the neurons and synapses. The user places neurons, and arbitrary weighted connections between the neurons, without particular regard for the organization of the network hardware. A router is used to convert the logical view to a physical representation that can be examined with the physical editor.

The network compiler converts the user-readable input file to an input file for the place-and-route program. Programming the network then proceeds in the same manner as with the logical editor.

Finally, network operation can be observed through the network state display. A 12-b A/D converter in the network controller can sample the state of the neurons through the OM line, as shown in Fig. 1. The sampling process is transparent to the operation of the neurons. The network state display supports two output modes. In mode 1, a strip chart is shown for each neuron. The magnitude of a neuron voltage at successive time slices is graphed in each chart. In mode 2, each neuron is represented by pixels whose intensity is proportional to the neuron output voltage.

Currently, all of the above systems are functional, except for the network compiler and the place-and-route systems, which are still in development. An 80386-based personal computer (PC with ISA bus) is used as the host computer. The network controller consists of a buffer memory and PAL-based finite state machine. Together, the network controller and 80386 PC can serially program the network at 2.5 Mb/s. Lastly, the network controller has random access read and write capability. Hence, any individual chip can be read or written without modifying the other chips in the system.

V. Performance and Results of a Prototype Network

A prototype analog neural system has been completely assembled and tested based on the units described above. As discussed in Section IV, software has also been written to control the configuration and monitor the network operation from the host computer. The prototype consists of 72 neurons, 2466 synapses, 21120 switches, and 36 time constants. Each neuron has 32 inputs. The system contains in total over 100 separate chips connected together as shown in Fig. 1. The total memory to digitally set the network configuration is about 40 kb, which can be downloaded in 16 ms. The system is realized on three processor boards. The current system is a test version of a larger one that is currently being developed and that will have over 1024 neurons with 64 inputs per neuron.

As mentioned earlier, the components and the overall system have been designed for real-world applications, in particular speech. As a result, the time response of the neurons has been optimized for such applications where response times in the tens of microseconds up to ten of

milliseconds range are required in order to be able to handle and process the incoming signals in real time. Although the network could run at higher speeds, it was deliberately slowed down. The overall computational power of the network comes not so much from the individual components but from the highly interconnected and parallel architecture and the spatio-temporal processing of the incoming signals. For this reason, it is not so meaningful to characterize the performance of the network by the speed of the individual components. It is more instructive to look at the overall response of the network for particular applications.

The network has been configured for a number of applications, including a "winner take all" net, an associative network, a neural integrator, and several circuits for the computation of time-domain pattern primitives [27]–[29]. The latter ones have been realized by using feedback and synaptic time constants, which generate dynamic activity patterns in the absence of external inputs. For the "winner take all" net, 16 neurons were fully interconnected with mutual inhibitory connections. Experiments for different inhibitory gains were performed. For inhibitory connections of 0.9 a clear "winner" emerged, while for inhibitory connections of 0.5 contrast enhancement of the input patterns was observed. The network settled within one time constant of the neurons. Also experiments to verify the fan-in and fan-out capabilities were performed in which the outputs of 32 neurons were routed through synapses with gain of 0.03 and summed together into one neuron, proving that noise was no problem even at these small gains.

Another application of the network is the real-time analysis of acoustical patterns. This network is programmed for the primary decomposition of acoustical patterns into primitives which are functions of energy, space (frequency), and time [30], as shown in Fig. 11. The analog input signals to the network come from a bank of high-Q bandpass filters (Q_{max} = 300 dB/octave), with characteristics similar to the ones found in the cochlea. In this example eight bandpass filters were used with the following center frequencies: 400, 600, 750, 900, 1100, 1700, 2500, and 4000 Hz. The primary neurons (layer 1) receive inputs from the bandpass filters. These neurons are mutually inhibiting in a center-surround fashion with spatially decaying gains. Lateral inhibition is applied to enhance the frequency tuning of the primary neurons. The next stage involves the ON and OFF neurons, which compute the temporal positive and negative derivative of the amplitudes. The ON units receive undelayed excitatory and delayed inhibitory inputs from the neurons in layer 1, whereas the OFF units receive delayed excitatory and undelayed inhibitory inputs. ON and OFF neurons are mutually inhibiting. The units in layer 3 are the complements of the ON and OFF units. They are normally ON via a bias indicated by the arrow in the figure and are inhibited by the activity of the neurons in layer 2. The last stage computes changes in formant frequency and their direction (local rise and fall of frequency, i.e., motion) through a combination of outputs of neurons of level 2 and 3, as is shown in Fig. 11.

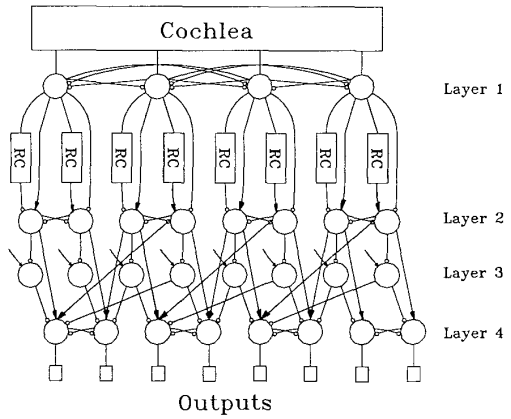

Fig. 11. Logical view of part of the network used for the primary decomposition of acoustical patterns. The neurons receive input from eight (only four are shown) bandpass filters.

The resulting outputs of the 56 neurons used in this experiment are given in Fig. 12 for a time period of 1000 ms. These waveforms are generated and recorded in real time. Fig. 12(a) gives the primary decomposition of an "ah" sound and Fig. 12(b) of a "dah" sound. Notice the ON and OFF units and their complementary outputs. The "dah" sound shows a formant transition at the lower frequencies (output of neuron 63) which is absent in the "ah" of Fig. 12(a). The network easily keeps up with the spoken word and decomposes the sound into its primitives in real time. These outputs can be used as input to a pattern recognition circuit to recognize individual phonemes. However, this is beyond the scope of this paper. The current network is too small to be useful for speech recognition. At least a 1000-neuron net will be required to perform all of the needed functions for speed recognition. Even for this simple prototype network (72 neurons) the speed advantage over digital simulations is impressive. A Sun 4/110 workstation was used to simulate the network, modeled as a set of differential equations, using a fourth-order Runge–Kutta algorithm [21]. We found that a digital computer with a speed of at least 10^{11} FLOPS is required to match the real-time performance of this neural network.

VI. Conclusions

The network described here is an attempt to construct a fully analog, parallel, and dynamic neural computer intended for real-world applications. Among its features are programmable synaptic time constants and weights over a large dynamic range. The performance of a prototype neural network, consisting of over 100 programmable chips, has been evaluated. The network has been used for several applications including real-time acoustical pattern analysis. The individual modules consisting of neurons, synapses, time constants, and analog switches are found to be functioning well within the overall network. The

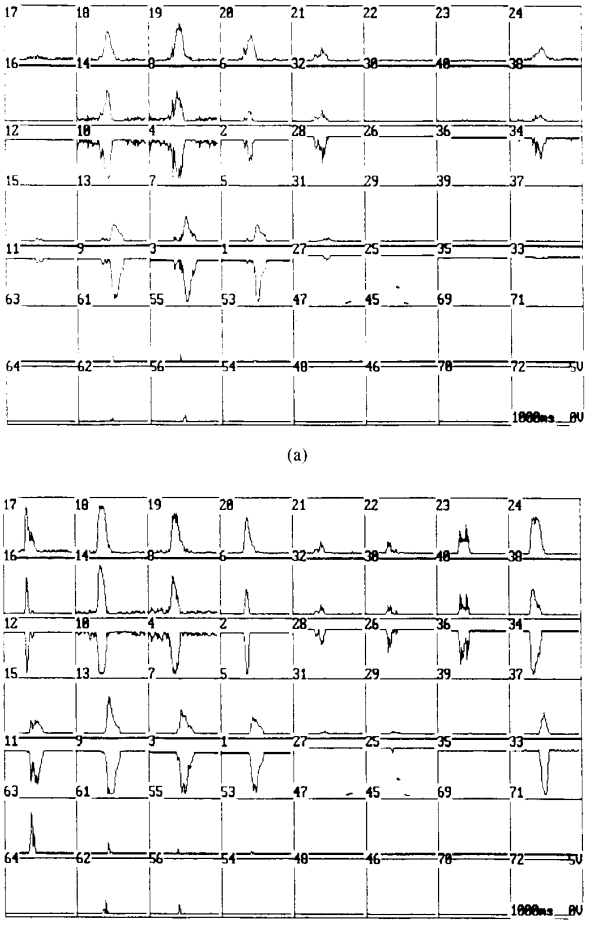

Fig. 12. Output voltages of 56 neurons during a 1000-ms time slice from a network configured for acoustical pattern analysis: (a) pattern for an "*ah*" sound, and (b) for a "*dah*" sound. Horizontal: 1000 ms; vertical: 5 V.

accuracy, noise levels, and stability proved to be entirely adequate, illustrating that scaled-up networks are feasible.

The main advantage of analog hardware realizations, working in continuous time, is the computational speed. Simulations on a digital computer show that a machine with a processing power of 10^{11} FLOPS is required in order to keep up with the neural computer. The next version, which will have 1024 neurons and 64 synaptic inputs per neuron, will have a performance in the order of 10^{12}–10^{13} equivalent FLOPS and is expected to be able to perform speech analysis in real time.

Acknowledgment

The authors would like to thank M. Massa and V. Agami for the support with the software development, S. Fernando for help with testing of the switch modules, and Prof. M. Steyaert for suggestions with the design of the time-constant circuits.

References

[1] T. Sejnowski, C. Koch, and P. S. Churchland, "Computational neuroscience," *Science*, vol. 241, pp. 1299–1306, 1988.

[2] S. Grossberg, *Studies of Mind and Brain*, vol. 70, R. Cohen and M. Wartofskey, Eds. Boston, MA: D. Reidel, 1982.

[3] J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Nat. Acad. Sci.*, vol. 81, pp. 3088–3092, 1984.

[4] D. E. Rumelhart and J. L. McClelland, Eds., *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, vols. 1, 2. Cambridge, MA: MIT Press, 1986.

[5] T. Kohonen, "Correlation matrix memories," *IEEE Trans. Comput.*, vol. C-21, pp. 353–359, 1972.

[6] P. Mueller, T. Martin, and F. Putzrath, "General principles of operations in neuron nets with application to acoustical pattern recognition," *Biological Prototypes and Synthetic Systems*, vol. 1. New York: Plenum, 1962, pp. 192–212.

[7] B. Widrow and M. E. Hoff, "Adaptive switching circuits," in *1960 IRE WESCON, Conv. Rec.* (New York), vol. IRE, 1960, pp. 96–104.

[8] L. E. Atlas and Y. Suzuki, "Digital systems for artificial neural networks," *IEEE Circuits Devices*, pp. 20–24, 1989.

[9] M. Ysunaga *et al.*, "Design, fabrication and evaluation of a 5-inch wafer scale neural network LSI composed of 576 digital neurons," in *Proc. IJCNN Int. Joint Conf. Neural Networks*, vol. II, 1990, pp. 527–535.

[10] R. Hecht-Nielsen, "Neurocomputing: Picking the human brain," *IEEE Spectrum*, pp. 36–41, in 1988.

[11] H. P. Graf and L. Jackel, "Analog electronic neural network circuit," *IEEE Circuits Devices Mag.*, pp. 44–49, July 1989.

[12] H. Graf and D. Henderson, "A reconfigurable CMOS neural network," in *ISSCC Dig. Tech. Papers*, 1990, pp. 144–145.

[13] B. Boser and E. Sackinger, "An analog neural network processor with programmable network topology," in *ISSCC Dig. Tech. Papers*, 1991, pp. 184–185.

[14] S. Satyanarayana, Y. Tsividis, and H. P. Graf, "A reconfigurable analog VLSI neural network chip," in *Advances in Neural Information Processing Systems 2*, D. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1990, pp. 758–768.

[15] J. Raffel, J. Mann, R. Berger, A. Soares, and S. Gilbert, "Electronic implementation of neuromorphic systems," in *Proc. IEEE Custom Int. Circuits Conf.*, 1988, pp. 10.1.1–10.1.7.

[16] J. Alspector and R. B. Allen, "A neuromorphic VLSI learning system," in *Advanced Research VLSI, Proc. 1987 Stanford Conf.*

[17] J. White *et al.*, "Parallel architecture of 2D Gaussian convolution of images," *Neural Networks*, vol. 1, p. 415, 1988.

[18] M. Holler, S. Tam, H. Castro, and R. Benson, "An electrically trainable artificial neural network etann with 10240 floating gate synapses," in *IJCNN Int. Joint Conf. Neural Networks*, vol. 2, 1989, pp. 191–196.

[19] A. Moopenn, T. Duong, and A. Thakoor, "Digital-analog hybrid synapse chips for electric neural networks," in *Advances in Neural Information Processing Systems, 2*, D. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1990, pp. 769–776.

[20] C. Mead, *Analog VLSI and Neural Systems*. Reading, MA: Addison-Wesley, 1989.

[21] P. Mueller *et al.*, "Design and fabrication of VLSI components for a general purpose analog neural computer," in *IEEE Workshop Analog VLSI Implementation of Neural Systems, Circuits Syst. Conf.* (Portland), C. Mead and M. Ismail, Eds. Boston: Kluwer, 1989.

[22] N. Farhat, "Optoelectronic neural networks and learning machines," *IEEE Circuits Devices*, pp. 32–41, 1989.

[23] K. Kyuma, "Optical architectures for neuron circuits," in *Proc. 1991 IEEE Int. Symp. Circuits Syst. (ISCAS91)*, vol. 3, pp. 1384–1387.

[24] P. Mueller and J. Lazzaro, "A machine for neural computation of acoustical patterns with application to real speech recognition," in *AIP Conf. Proc.*, vol. 151, 1986, pp. 321–326.

[25] K. Bult and H. Wallinga, "A CMOS four-quadrant analog multiplier," *IEEE J. Solid-State Circuits*, vol. SC-21, no. 3, pp. 430–435, June 1986.

[26] M. Steyaert, P. Kinget, W. Sansen, and J. Van der Spiegel, "Full integration of extremely large time constants in CMOS," *Electron. Lett.*, vol. 27, no. 10, pp. 790–791, 1991.

[27] P. Mueller, *et al.*, "Design and performance of a prototype general purpose analog neural computer," in *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, vol. I, 1991, pp. 463–468.

[28] J. Van der Spiegel *et al.*, "A multichip analog neural networks," in

Proc. Int. Symp. VLSI Tech. Syst. Applications, VLSITSA, 1991, pp. 64–68.

[29] P. Mueller *et al.*, "Design and performance of a prototype analog neural computer," in *Proc. 2nd DOD Workshop Neural Networks* (Huntsville, AL), Sept. 1991.

[30] P. Mueller, "Computation of pattern primitives in a neural network of acoustical pattern recognition," in *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, vol. I, 1990, pp. 149–151.

Jan Van der Spiegel (S'73–M'79–SM'90) was born in Aalst, Belgium. He received the engineering degree in electromechanical engineering, and the Ph.D. degree in electrical engineering from the Katholieke University of Leuven, Belgium, in 1974 and 1979, respectively.

From 1980 to 1981 he was a postdoctoral fellow at the University of Pennsylvania, Philadelphia, where he became an Assistant Professor of Electrical Engineering in 1981. Currently, he is an Associate Professor of Electrical Engineering and of Material Science and Engineering. He holds the Bicentennial Chair of the Class of 1940 and is Director of the Center for Sensor Technologies at the University of Pennsylvania. His research interests are in artificial neural networks, CCD sensors, microsensor technology, and low-noise, low-power, and high-speed analog integrated circuits.

Dr. Van der Spiegel is the recipient of the Presidential Young Investigator Award, an IBM Award for Young Faculty Development, the Reid S. Warren Award, and the C.M. Lindback Award for distinguished teaching. He is Editor for *Sensors and Actuators* (North and South America).

Paul Mueller received the M.D. degree from Bonn University, Germany.

He has worked in molecular and systems neuroscience since 1953 and has been involved in theoretical studies and hardware implementation of neural networks since the early sixties.

David Blackman received the B.S. degree in computer science and engineering from the University of Pennsylvania, Philadelphia, in 1981, and the M.S. degree in computer science from Rochester Institute of Technology in 1986.

From 1981 to 1986, he worked as an Engineer at Xerox Corporation in Rochester, NY. He is currently a graduate student in the Department of Neuroscience at the University of Pennsylvania where he is working on models of the visual cortex and VLSI implementation of neural networks.

Peter Chance received the B.S. and M.S. degrees in computer science from the University of Pennsylvania, Philadelphia.

From 1974 to 1984 he worked in robotics and machine learning. His current interests include speech recognition and robotic control using neural networks.

Chrostopher Donham was born in Greenwich, CT, in 1967. He received the B.S.E. and M.S.E. degrees in electrical engineering from the University of Pennsylvania, Philadelphia, in 1989 and 1991, respectively. He is currently working towards the Ph.D. degree in electrical engineering at the University of Pennsylvania where he is researching neural network based speech recognition.

Mr. Donham received the Outstanding Student Member Award for the Philadelphia branch of the IEEE in 1988.

Ralph Etienne-Cummings was born in Victoria, Seychelles, in 1967. In 1988 he graduated Valedictorian and summa cum laude from Lincoln University, PA, with the B.Sc. degree in physics. He is currently pursuing the Ph.D. degree in electrical engineering at the University of Pennsylvania, Philadelphia. His research interests include hardware implementation of neural networks and visual tracking systems.

Mr. Etienne-Cummings was a Fountain Fellow from 1988 to 1990 and is currently a Harris Fellow.

Peter Kinget (S'88) was born in Buffalo, NY, on June 24, 1967. He received the M.Sc. degree in electrical and mechanical engineering in 1990 from the Katholieke Universiteit of Leuven, Belgium. During the summer of 1989 he participated in the summer research program on sensor technologies for undergraduate students at the University of Pennsylvania, Philadelphia. Currently, he has a N.F.W.O. fellowship which allows him to work as a Research Assistant at the ESAT Laboratory, Katholieke Universiteit Leuven. He is working towards the Ph.D. degree on the analog VLSI implementation of neural networks. His research interests include neural networks and analog VLSI design.