

AN ABSTRACT OF THE THESIS OF

James Hansen for the degree of Master of Science in Electrical and Computer Engineering presented on June 30, 1989.

Title: A Time-Multiplexed Switched-Capacitor Circuit For Neural Network Applications

Abstract approved: *Redacted for Privacy*

David J. Allstot

Analog computation in the form of neural networks is currently receiving much attention. Existing algorithms cannot be easily implemented in hardware because of the large number of neurons needed and the number of connections necessary between them. These problems have motivated development of alternatives to a conventional implementation. Hence, a time-multiplexed switched-capacitor computational block with some neural network characteristics has been developed. The goal of this work is not to exactly model biological neuron qualities but rather to emulate their primary characteristics to the extent necessary to solve some types of problems not suited to digital processing.

A Time-Multiplexed Switched-Capacitor Circuit For
Neural Network Applications

by

James Hansen

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Completed June 30, 1989

Commencement June 1990

APPROVED:

Redacted for Privacy

Professor of Electrical & Computer Engineering in charge of major

Redacted for Privacy

Head of Department of Electrical & Computer Engineering

Redacted for Privacy

Dean of C

Date thesis is presented June 30, 1989

ACKNOWLEDGEMENT

I deeply appreciate the contributions of David Allstot, who had the initial idea of this circuit and provided advice through all stages of development.

This work would not have been successful without the effort of Jeanette Skelton, who devised the algorithm and programmed the controller to test the circuit.

I would also like to thank Dan Shum for valuable assistance with op-amp design.

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	CIRCUIT OPERATION	4
	Initialization	5
	Sequential Sample and Update	6
	Control Circuitry	7
III.	LIMITATIONS	9
	Parasitic Capacitance	9
	Switch Current Leakage	10
	Op-amp Response	11
	Clock Skew	12
IV.	BREADBOARD CIRCUIT	13
V.	INTEGRATED CIRCUIT	16
	Multiplying Digital to Analog Converter (MDAC)	16
	Decoder	17
	Synchronization register	17
	Storage cell (neuron)	18
	Op-amps	19
	Analog Mux	19
	Verification	20
	DRC	20
	LVS	21
	Possible improvements	21
VI.	APPLICATIONS	22
	Hopfield Associative Memory	22
	Other Potential Applications	23
VII.	CONCLUSION	25
VIII.	REFERENCES	57
IX.	APPENDICES	
	1. Parasitic capacitance equation derivation	59
	2. SPICE parameters	61
	3. DRC command file	62
	4. LVS command file	66

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
1.	Switched-capacitor integrator	26
2.	4 bit MDAC	26
3.	Possible weighting factors	27
4.	Programmable time multiplexed integrator	27
5.	Example control signals	28
6.	Sequential sample and update circuitry	29
7.	Complete circuit	30
8.	Parasitic capacitance model	31
9.	T-gate parasitic capacitance model	31
10.	Breadboarded circuit layout	33
11.	Output LED's	34
12.	Half wave rectifier, comparator, and serial-parallel conversion	35
13.	Integrator, buffer, and sample/hold	36
14.	Digital control circuitry	37
15.	Integrated circuit block diagram	38
16.	MDAC layout	40
17.	Decoder input stage and 8 output stages	41
18.	Static 6-64 decoder floorplan	42
19.	Synchronization latch	43
20.	Latch transistor diagram	44
21.	Neuron access circuitry	45

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
22.	Layout of neuron access circuitry	46
23 (a).	Op-amp block diagram	47
23 (b).	Op-amp biasing circuitry	48
23 (c).	Op-amp input stage	49
23 (d).	Op-amp output stage	50
24.	Op-amp layout	51
25.	Integrator output relative to neuron access (SPICE)	52
26.	Op-amp open loop frequency response	53
27.	Analog multiplexor	54
28.	DRC rules	55
29.	Improved neuron access circuitry	56

LIST OF TABLES

<u>Table</u>		<u>Page</u>
1.	List of parts for breadboard circuit	32
2.	Integrated circuit pinout	39

A TIME-MULTIPLEXED SWITCHED-CAPACITOR CIRCUIT FOR NEURAL NETWORK APPLICATIONS

I. INTRODUCTION

Two problems encountered with silicon implementation of neural networks are realizing a useful number of "neurons" within an acceptable chip area and providing sufficient interconnection of these neurons. The brain contains 10^{10} to 10^{11} neurons, and each neuron is connected to 10^3 to 10^5 other neurons [1]. Although computation can be done with fewer neurons, substantial numbers of fully connected neurons are needed to perform useful functions. Typically, n neurons require n op-amps [2]. Also, since all neurons are active simultaneously, each of the n^2 interconnections must be effected with a physical line. This approach has been implemented with several hundred neurons, but requires special fabrication processes, and the number of neurons is limited by the area of the interconnection matrix [3]. An alternative approach is to use capacitors to model neurons and share the required amplifying circuitry, such that one and only one capacitor is active at once, thereby requiring only one op-amp and reducing the number of interconnections to n . Since gates in present-day silicon technology switch roughly 10^6 times faster than their biological counterparts, the time-sharing approach is feasible for application to problems requiring real-time solutions, such as

recognition and image processing. Time-multiplexing of silicon neural network circuits has been suggested in [4] and [5], among others. Time-multiplexing makes full interconnection easily attainable, at the expense of reduced fault tolerance. Furthermore, multiplexing makes more efficient use of relatively expensive interconnect lines, since these lines are inactive most of the time in non-multiplexed applications.

The circuit described in this thesis is intended to serve as a co-processor to a digital microprocessor. The microprocessor specifies sample and update sequences, interconnection, and weighting factors. Weighting factor adjustment associated with learning is therefore under microprocessor control. A major advantage to this approach is the fact that connectivity is completely programmable. Internal data processing is analog, making the circuit a hybrid of digital and analog processing.

The full interconnection possible within this circuit could allow the circuit to function as a neural network "layer," with layers connected through multiple circuits. Functionality has been demonstrated with circuit level simulation and also with a breadboard circuit applying an associative memory algorithm using weights of -1, 0, and 1. Additionally, an integrated version of the circuit has been developed, using a 2 micron double-metal double-poly p-well process from MOS Implementation Services (MOSIS). Although associative memory can easily be implemented with purely digital hardware, the algorithm is useful for demonstrating the analog processing capabilities of the circuit. This thesis will describe circuit operation, discuss limitations imposed by non-ideal effects, and outline potential applications. General theory will be presented first, followed by specific details of the breadboarded circuit

and then of the integrated circuit. Throughout this thesis, the following definitions/conventions are used:

state = voltage on capacitor C_{ni} at the end of an update sequence

T-gate = parallel-connected n and p pass transistors

V_T = transistor threshold voltage

neuron = storage capacitor to emulate biological neuron

op-amp = operational amplifier

mil = .001 inch

standard cell = a simple circuit building block which employs a
standardized set of wiring rules

route-through = signal passing through a standard cell unaffected

II. CIRCUIT OPERATION

Individual biological neurons sum weighted inputs to establish their excitation state. These inputs can be either negative or positive, for inhibitory or excitatory action, respectively. Electrically, this summation can be realized with a conventional integrator, or a switched-capacitor integrator. The operation of the switched-capacitor integrator of Figure 1 has been shown to be unaffected by parasitic capacitances on nodes W, X, and Y [6, p.277] for input V_{in} as shown. For circuit operation with clock phases shown without parentheses, the output voltage can be shown to be

$$V_{out} = -C_1/C_2 V_{in}(t-1) + V_{out}(t-1) \quad (1)$$

In other words, the new output voltage is a weighted sample of the input voltage subtracted from the previous output voltage. For clock phases as shown inside the parentheses,

$$V_{out} = C_1/C_2 V_{in}(t-1) + V_{out}(t-1) \quad (2)$$

or a weighted sample of the input voltage added to the previous V_{out} .

Consider the circuit of Figure 2. This capacitive weighting array replaces C_1 in Figure 1, thereby allowing multiple possible weighting factors of V_{in} in equations (1) and (2). The C_o capacitor to ground and the switched capacitors 1, 2, 4, and 8 C_o make this array binary weighted, which results in a uniform distribution of possible weighting factors, as shown in Figure 3. This type of circuit is commonly known as a multiplying digital to analog converter (MDAC), and allows connection

weights to be programmable. For this circuit, the capacitance of the entire array was chosen equal to C_2 , or $C_2 = 16C_o$. Figure 4 presents the modified integrator. Reference [7] proposed the binary weighted capacitive array to emulate variable connection strengths, and [8] replaced the inverter with an op-amp to obtain linear processing capabilities. The novelty of the Figure 4 circuit is the multiple capacitors C_{ni} as storage elements, with element values being represented by the corresponding voltage across that capacitor. Access to each C_{ni} is time multiplexed. In effect, nodes Y and V_{out} can be considered the analog equivalent of a microprocessor data bus, since many elements share these nodes. Charge is conserved on any C_{ni} while n_i is open and therefore the neuron state does not change, regardless of events on nodes Y and V_{out} . Data processing requires two distinct modes of operation: (1) initialization, and (2) sequential sample and update. These modes and control circuitry to implement these modes will be described.

Initialization

Storage elements C_{ni} must be activated to their beginning values before processing can occur. Initialization is accomplished by sequentially accessing each C_{ni} in phase with ϕ_2 while presenting the desired voltage to the V_{in} node during ϕ_1 . In the integrated circuit, the binary weighted capacitive array at the integrator input acts as a 4 bit digital to analog converter. This array and the op-amp saturation limits are sufficient to approximate continuous, non-linear activation functions required for analog parallel distributed processing [1, p. 47]. A resolution of 4 bits is

generally believed to be sufficient to model biological neural processing [9]. Figure 5 shows control signals for initialization of C_{ni} , $i = 0 - 3$, with arbitrarily chosen V_{in} values. As evident from this figure, during initialization feedback switch SH is exactly out of phase with ϕ_2 , to maintain op-amp feedback. In this mode, the integrating op-amp is merely transferring charge stored on the MDAC onto the respective neurons.

Sequential Sample and Update

A processing cycle consists of a non-destructive sample phase (ϕ_1) and an update phase (ϕ_2), whereby V_{nj} is altered by an amount proportional to V_{ni} , while leaving V_{ni} unchanged. A complete sequence for update of all C_{ni} requires $n^2 - n$ cycles (assuming no C_{ni} is sampled and updated to itself). This situation represents full interconnection within the network. If less than full interconnection is required, the time for an update sequence can be reduced, since interconnect is realized in time instead of hardware. Consider now the addition of sample and hold circuitry to temporarily store neuron values, as shown in Figure 6. During the ϕ_1 phase, the integrator output voltage is driven onto sample and hold circuitry. This voltage is held unaltered during the ϕ_2 phase. The sample and hold allows the voltage of each sampled C_{ni} to be presented as the integration input voltage to each updated C_{ni} during either ϕ_1 or ϕ_2 (and therefore subtraction or addition). The input capacitive array now allows for a range of weighting factors w , $-15/16 < w < 15/16$. Figure 5 shows control signals for a 4 C_{ni} update, implying the equations:

$$n_0(t) = 15/16n_1(t-1) + 15/16n_2(t-1) - 15/16n_3(t-1) + n_0(t-1)$$

$$n_1(t) = 15/16n_0(t) - 15/16n_2(t-1) + 15/16n_3(t-1) + n_1(t-1)$$

$$n_2(t) = 15/16n_0(t) - 15/16n_1(t) + 15/16n_3(t-1) + n_2(t-1)$$

$$n_3(t) = -15/16n_0(t) + 15/16n_1(t) - 15/16n_2(t) + n_3(t-1)$$

Note that time multiplexing results in storage element values being a function of both present and previous states, depending on the update order. Sample and update order is specified by the micro-controller. This approach is reasonable, since most neural networks assume asynchronous update.

In practice, the parasitic capacitance C_{out} develops significant charge during the update phase. This charge is discharged into the op-amp output through SH prior to the subsequent sample phase.

Control Circuitry

At the end of each C_{ni} update, the output of the integrator is compared to an arbitrary threshold voltage, which for applications explored so far has been $VDD/2$. The comparator output (digital) is fed into a n -bit serial-to-parallel shift register, where n is the total number of C_{ni} . Figure 7 presents the complete circuit. Note that since only one C_{ni} is active at any one time, a decoder can be used to reduce the necessary C_{ni} input control signals from n to $\log_2 n$ plus an enabling signal, which is required to disable all C_{ni} when SH is active.

Boundaries of the integrated circuit are shown by the dotted line. An analog output from the processing stage is included to allow multiple circuits to be connected together, but has not yet been utilized.

Ensuring that the sample of C_{ni} is precisely out of phase with the update of C_{nj} is particularly critical, since overlap during the high clock state would allow charge transfer between C_{ni} and overlap during the low clock state would cause the op-amp to lose feedback. Therefore, a parallel-in/parallel-out shift register was used to synchronize control signals, as shown in Figure 7.

III. LIMITATIONS

The feasibility of this circuit for many applications can be measured by the number of neurons that can be implemented without degrading computational capabilities. Circuit performance is limited by four separate non-ideal effects: parasitic capacitance, switch current leakage, op-amp response, and clock skew. Each effect limits a different aspect of circuit operation, and will be considered independently. Since processing requirements for different applications vary widely, it is difficult to quantify limiting factors exactly. The equations presented are based only on first-order effects, to serve as an estimate of circuit limitations.

Parasitic Capacitance

Parasitic capacitances exist between any nodes which are physically in close proximity. These parasitics need to be considered in the circuit sections associated with the neuron capacitors. The model of Figure 8 is used to analyze effects of coupling and clock feedthrough capacitances. For a sample of C_{ni} and update of C_{nj} , conservation of charge analysis implies that C_{i3} degrades the voltage V_{ni} by an amount

$$\Delta V_{nc} = [C_{i3} / (C_{i3} + C_{ni})] (V_{out2} - V_{out1}) \quad (3)$$

where V_{out1} and V_{out2} are sample and update output voltages, respectively and ΔV_{nc} is the change in neuron voltage due to parasitic capacitance. In the breadboarded circuit, C_{i3} is the pin to pin package

capacitance, and in the integrated circuit, C_{i3} is the equivalent capacitance of $C_{source-bulk}$ in series with $C_{bulk-drain}$. Effects of parasitic capacitance on the output node are reduced by discharging this capacitance through feedback switch SH after every update cycle.

Likewise, for a sample of C_{ni} ,

$$\Delta V_{nc} = [-C_{i1} / (C_{i1} + C_{ni})] (V_{DD} - V_{SS}) \quad (4)$$

Using a T-gate for the switch provides some cancellation of this parasitic effect. Figure 9 is used to derive the modified equation

$$\Delta V_{nc} = [(C_{i4} - C_{i1}) / (C_{i4} + C_{i1} + C_{ni})] (V_{DD} - V_{SS}) \quad (5)$$

However, since for the integrated circuit C_{i1} and C_{i4} are dominated by the gate overlap capacitances C_{gson} and C_{gsop} and the width of the p device is roughly twice that of the n device, significant non-ideal effects remain. Appendix 1 presents the derivation of equations (4) and (5). Using the T-gate also allows full voltage levels to be passed, whereas a single n or p pass transistor would limit the output swing to $V_{DD} - V_{SS} - V_T$. Assuming $C_n \gg C_{i4}$, C_{i5} the total voltage degradation becomes,

$$\Delta V_{ntotal} = (n)(C_{i4} - C_{i1})(1/C_{ni})(V_{DD} - V_{SS})$$

or

$$n = [(\Delta V_{ntotal}) / (V_{DD} - V_{SS})] (C_{ni}) / (C_{i4} - C_{i1}) \quad (6)$$

Switch Current Leakage

Even while a particular neuron is not active, charge can leak through the associated open switch. For a nominal reverse leakage current density of 10 pA/mil² [6, p. 472] and a source diffusion area of 50 μm^2 , the

leakage current I_l is 0.8 pA. If the maximum acceptable error voltage is 50 mV, then from

$$q = I_l t = cv$$

$$t/C_n = V/I_l = 6.3 \times 10^{10} \text{ sec/f}$$

where t is total time for an update sequence. This relation indicates the tradeoff between t and C_n - a longer time t requires a larger neuron capacitance C_n . $C_n = 10$ pF therefore limits t to .63 sec. Since

$$t = (n^2 - n) T, \quad (7)$$

where T = cycle period, leakage current limits n , the number of C_{ni} .

Op-amp Response

For applications requiring real-time processing (i.e., the complete network must respond as fast as a biological network), the number of neurons may be limited by how fast the op-amps can drive their capacitive loads. Both the integrating op-amp and the buffering op-amp (not shown) must be able to drive the output voltage from $V_{out}(\min)$ to $V_{out}(\max)$ in $T/2$. This limitation sets the minimum cycle period T . For the breadboard circuit, op-amp slew rate limits T . The integrated circuit uses a non-slewing class-AB structure, so T is limited by the op-amp bandwidth. From (7), it is seen that t is directly proportional to T . Assuming $t = 10$ milliseconds (roughly the processing speed of a biological neuron) and $T = .1$ microseconds results in $n = (t/T)^{1/2} = 300$.

Clock Skew

Although a register is used to synchronize neuron control signals, using a CMOS T-gate requires inverting each control signal for one side of the T-gate. The small delay associated with this inversion results in a brief time when two different neuron switches are both on, allowing charge to transfer directly from one neuron capacitor to another. To a first order, the resulting voltage degradation can be estimated by modeling the switch (on) as a resistor R .

$$\Delta V_{ns} = (\Delta q_s/C_n) = (i_s)(T_s)(1/C_n) = (V_{n1} - V_{n2})(1/R)(T_s)(1/C_n)$$

where T_s is the clock overlap time. Assuming $V_{n1} - V_{n2} = 5$, $R = 10 \text{ K}\Omega$, $C_n = 10 \text{ pF}$, $T_s = .5 \text{ ns}$ yields a $\Delta V_{ns} = .025 \text{ volt}$. Total voltage degradation is

$$\Delta V_{n\text{total}} = (\Delta V_{ns})(2)(n - 1) \quad (8)$$

since each neuron is accessed $n - 1$ times for sampling and $n - 1$ times for updating. Using (8) and setting $\Delta V_{n\text{total}} = 5$ would appear to limit n to about 50. However, as will be discussed in the Integrated Circuit section, certain improvements can be made to increase the maximum n to 500 - 1000.

IV. BREADBOARD CIRCUIT

Because of the considerable time and effort involved with developing an integrated circuit, a breadboard circuit was a useful method of more easily verifying circuit functionality. The circuit simulator SPICE was used to analyze small sections of the circuit, but run-time limitations made it impractical for analyzing the complete circuit. Discrete digital and analog components were used to implement the circuit of Figure 4. Connections were made with 30 gauge wire-wrap. Although more tedious to implement, wire-wrapping is superior to standard plug-in breadboards in both reliability and stray capacitances. In the breadboarded circuit, stray capacitances limited the MDAC to 1 bit, thereby allowing weights of -1, 0, and 1. A total of 16 mica capacitors of 270 pF were used as neurons. Capacitors with mica as the dielectric have a high quality factor Q , meaning very low charge leakage, required for slow circuit operation. Table 1 lists the discrete components required. Reference [10] gives electrical specifications for the switches and op-amps. Supply voltages were ± 12 volts for analog components and +5 and 0 volts for digital components. Functionality was demonstrated with an auto-associative memory algorithm (to be described in the applications section).

Since the analog circuitry directly drives the digital comparator, extra circuitry was necessary to establish voltage level compatibility. A simple half-wave inverting rectifier with supply voltages of +5, -12 and a ground reference voltage solved this problem, at the expense of clipping all

all neuron voltage values above 0. This limitation was acceptable for the algorithm implemented.

Unfortunately, the parasitic capacitance C_{i3} of Figure 8 was a relatively large 3 pF. This large value was due to the fact that the signal input pin is adjacent to the signal output pin in the switch component (LF13331). Adjacent pins on an IC typically have a coupling capacitance of 3 - 4 pF. Since V_{out} changes twice every cycle, or $2(n^2 - n) = 480$ times in a complete sequence, effects of C_{i3} and C_{j3} tend to dominate, limiting the total number of updates and minimum C_n . Using equation (3), this value of C_{i3} limits the total number of switches to about 50, assuming that the neuron voltage is allowed to degrade to 1/2 of its initial value and that the output voltage is driven between the supply rails at each switch. However, this last assumption is not realistic, and in fact good success was obtained with complete interconnection (requiring 240 sample/update cycles).

Figure 10 presents a zone by zone layout diagram of the breadboarded circuit. Figures 11 through 14 show the functional circuitry within the zones. Inputs for circuit control are accessible through the 34 pin connector of zone F. These inputs were driven by a Motorola 6809[®] based controller board, which in turn was loaded from an IBM PC-AT[®] compatible computer. Connection from the computer to the controller board was with standard RS-232 cabling, and connection from the controller board to the circuit board used ribbon cable. The speed of the controller board limited a sample/update cycle to about 15 μ sec. See [11] for a detailed description of circuit control.

Light-emitting diodes (LED's) were used at the output as a simple but effective means of displaying latched values of the serial to parallel shift register. Resistors were included to provide current-limiting protection to the LED's.

V. INTEGRATED CIRCUIT

While the breadboard circuit was useful for proving basic functionality, the limits imposed by parasitic capacitance required circuit integration to realize a larger number of neurons. A block-diagram plot of the Very Large Scale Integration (VLSI) circuit is shown in Figure 15 and Table 2 lists the associated pinout. The circuit has a die size of 2230 X 2250 microns, and a core size of 1800 X 1800 microns. Although the minimum polysilicon feature size is 2 microns for this process, all transistor gates are at least 3 microns long, to allow operation at voltages higher than 5 volts. The chip fabrication will be done by Orbit Semiconductor[®], through MOSIS. Circuit development consisted of three phases: 1) simulation, 2) layout, and 3) verification. All three phases were completed using Mentor Graphics[®] software running on an Apollo[®] computer system. SPICE parameters for simulation were obtained by averaging measured parameters from two previous wafer lots from Orbit[®]. These parameters are documented in Appendix 2. A block-by-block description of the VLSI circuit and verification procedures follows.

Multiplying Digital to Analog Converter (MDAC)

A circuit commonly employed to multiply an analog signal by a digital (control) signal is detailed in Figure 2. The transfer function for this circuit is

$$|V_{\text{out}} / V_{\text{in}}| = (\sum C_i) / (16 C_u),$$

where $\sum C_i$ is the sum of the accessed capacitors. Note that

$$|V_{\text{out}} / V_{\text{in}}|_{\text{max}} = 15/16.$$

Layout of the MDAC is shown in Figure 16. Capacitors are laid out as sums of unit capacitors in order to minimize effects from process variations. In other words, variations in dielectric thickness, mask alignment, and lateral etching should affect all capacitors roughly equally since the capacitors are composed of sums of identical blocks.

Decoder

A static-logic 6-64 decoder was implemented with standard cell inverters and 4 input NAND gates, as shown by the gate level diagram of the input stage and 8 of the total 64 output stages of Figure 17. Note that the circuit includes enabling signal G. Figure 18 shows a floorplan of cell arrangement within the decoder. All 64 output signals are on the bottom of the decoder block of Figure 15. This static logic approach was used for simplicity, but is very area inefficient.

Synchronization register

The simple gating latch of Figure 19 was used to synchronize the control signals. As evident from the transistor level diagram of Figure 20, clocked inverters were used instead of T-gates to drive the storage node. Large double inverters were used as clock drivers for this register latch. Latches associated with the 64 neuron control lines were laid out in

a 2 X 32 array, and 8 other latches for miscellaneous control signals were placed adjacent to this array.

Storage cell (neuron)

Circuitry for individual capacitive neurons and access switches is shown in Figure 21. Note that the switch is a fully complementary T-gate. Figure 22 presents the layout. Neuron capacitance with linear C-V characteristics is obtained in the region where poly1 overlaps poly2. For this process, typical poly1 to poly2 capacitance is $.5 \text{ fF}/\mu\text{m}^2$ [12], so an overlap area of $8000 \mu\text{m}^2$ results in a neuron capacitor of 4 pF. The 64 neurons were laid out in a 4 X 16 array. Since all control signals come from the register above, the first neuron row included 3 route-throughs/neuron, the second row 2 route-throughs/neuron, and the third row 1 route-through/neuron. These control signals were routed away from the capacitor. This routing is not required by MOSIS design rules [12], but done to minimize parasitic capacitive coupling. Ideally, the capacitors would be arrayed as sums of unit capacitors, as in the MDAC, but area constraints precluded that approach. However, the primary consideration for accuracy is relative capacitor ratios and not absolute values, so sufficient accuracy is achieved since all of the neuron capacitors are the same size.

Op-amps

All four op-amps use a class AB non-slewing structure, based directly on work presented in [13]. The variable tail current and adaptively biased output stages make this structure well suited for driving capacitive loads with minimal settling time. Development was done in three stages: 1) biasing, 2) differential input, and 3) output circuitry. Figures 23 and 24 show the op-amp schematic and layout. Figure 25 gives the transient SPICE output for a typical sample-update cycle. A load capacitor $C_L = 10$ pF was included to prevent oscillation. Phase margin is defined as the phase difference between the input signal and output response at the unity-gain frequency. Normally, a phase margin of 45° is sufficient to guarantee stability. Figure 26 presents the frequency response of this op-amp. Note that high gain is achieved well above 1 MHz, so a sample/update cycle time of 2 μ seconds is practical. Open-loop small signal gain is about 12,000 V/V.

Analog Mux

The analog multiplexor was implemented with 2 switches controlled by signals of opposite polarity, so that only one signal path is active at a time. Figure 27 presents transistor schematic for the analog multiplexor.

Verification

The large amount of data involved with IC layout makes some sort of automated checking almost mandatory. Two basic programs for data checking were employed, both part of the DRACULA® software package developed by ECAD®. Shell routines for calling these programs are included in the Mentor Graphics® software package, and were run as each block was completed and then used to verify integrity of the complete chip. The first type was a Design Rule Check (DRC) and the second was a Layout versus Schematic (LVS). A description of each type of checking follows.

DRC

This program does layer by layer checks of the layout to ensure that appropriate spaces and overlaps of each layer with respect to other layers has been allowed. The actual spaces and overlaps required depend on how masks are generated, the alignment of these masks, and details of the IC processing itself. Therefore, a different technology file must be generated for each process that an IC is to be fabricated in. The MOSIS design rules [12] are intended to be scalable so future technology files can be easily generated by simply multiplying each number by a scaling factor. Figure 28 shows a graphical representation of the design rules for this particular process, and Appendix 3 presents the DRACULA® command file to implement these rules.

LVS

All simulations are based on circuit schematics, so a comparison of layout circuitry and schematics is essential. The program extracts circuit information and connectivity data from the layout and compares this data to the schematic, so this type of checking is especially useful for finding errors in routing and functionality. Circuit elements and connectivity are defined in the LVS DRACULA® command file, which is presented in Appendix 4. Note that this file is technology dependent only for the names of the layers involved, which allows easier transfer to new technologies than DRC.

Possible improvements

Since the main focus of circuit integration was to prove feasibility, not optimize performance, certain improvements could be made. As evident in Figure 15, the decoder consumes a significant fraction of the chip area. Area efficiency could be improved by using dynamic logic for this part of the circuit. Effects of clock skew could be minimized by adopting the circuitry of Figure 29. By sizing inverters Q1 - Q3 appropriately larger than Q4 - Q5, clock overlap could be reduced. Finally, increasing chip size from $(90 \text{ mil})^2$ to a realistic $(300 \text{ mil})^2$ would allow placing substantially more neurons on a chip. Conservatively assuming the same neuron density of $85 \text{ mil}^2/\text{neuron}$ would result in over a 1000 neurons/chip.

VI. APPLICATIONS

Hopfield Associative Memory

The specific algorithm tested on the breadboard circuit was the Hopfield associative memory algorithm [14]. This algorithm is well suited to the limitations imposed by the breadboarded circuit, namely poor weight resolution and a limited number of neurons. In this algorithm the network of n neurons stores m n -bit words, or memories, in the symmetric weights between the neurons. The weight between two neurons i and j , W_{ij} , is determined by bits i and j of each of the memories. If the bits are the same, 1 is added to W_{ij} , and if they are different, 1 is subtracted from W_{ij} . Thus if bits i and j are the same in all m memories W_{ij} will be m . To look up a memory, a key is used which is sufficiently close to one of the memories and the neurons are initialized to the corresponding bits of the key. For a neuron i , the output V_i is determined by

$$\sum W_{ij} V_j > \theta_i \Rightarrow V_i = 1$$

$$\sum W_{ij} V_j < \theta_i \Rightarrow V_i = 0$$

where θ_i is the threshold of the neuron. After a few updates of the network, the neurons will usually converge to the memory which is closest to the key. Hopfield found that for optimal recall, m should be less than $0.15 n$.

Since $(16) (.15) = 2.4$, the breadboarded circuit was tested with two randomly generated memories. The weights, which for two memories

must be -2, 0, or 2, were normalized by dividing them by two, so that they fell between -1 and 1. Although the circuit performed well, it has several differences from the Hopfield model. The capacitors which represent the neurons store the sums of the products of weights and inputs, or $\sum W_{ij} V_j$, not the state of the neurons, V_i . Therefore, inputs to the neuron may not be strictly 0 or 1, but range between ± 2 (because the op-amp saturates at ± 10 V, and 5 V was chosen to represent a 1). This greater range results in an amplification of the connections between neurons, but does not usually lead to an incorrect state. The other difference arises because of the saturation of the op-amp. Since the summation of the weights and inputs to a neuron may result in a level greater than 2 or less than -2, this sometimes leads to results which differ from those of the Hopfield model. However, any physical circuit would have this limitation. See reference [11] for a more complete description of how this algorithm was implemented.

In all, 16 test cases were run. For 15 of these tests, all 16 neurons converged to the expected state. In the remaining test, the sole incorrect neuron was updated late in the sequence, after significant cumulative effect from equation (3). Therefore, it was concluded that the circuit was functioning properly.

Other Potential Applications

The circuit may be used to implement more complex models also. Since the weights are stored in the controlling microprocessor, they may easily be changed during circuit operation in order to realize learning

algorithms. The variable threshold may be used to change the threshold function.

The factor limiting applications with the breadboarded circuit was the number of neurons. For example, the well-known traveling salesman problem requires k^2 neurons to solve an k city problem, and the number of unique solutions is $(1/2)(k-1)!$. A circuit with 16 neurons can therefore only solve a 4 city problem, which has only 3 possible solutions. However, the 64 neuron integrated circuit could be used to solve a much more meaningful 8 city problem.

For multi-stage networks, the total number of neurons in the network may well exceed n , the number of neurons on one chip. Since within a chip, neurons may be easily connected, but from chip to chip neuron interconnection is more difficult, circuit partitioning should minimize chip to chip interconnection. If the number of neurons in the first and second stages is less than n , the network may be divided between two chips. The first chip could implement the first and second stages, and the second chip could implement the second and third stages. The analog outputs of the second stage neurons of the first chip could be used to initialize the second stage neurons of the third stage. Thus the circuit could be implemented at the cost of duplicating the second stage neurons.

VII. CONCLUSION

A simple neural network circuit combining the concept of time multiplexing with a switched-capacitor integrator approach has been presented. Standard CMOS processing techniques can be used to implement this circuit, which is a hybrid of digital and analog processing. Benefits of this circuit include multiple storage elements for each integrating amplifier and the allowance for full interconnection between elements, implemented in time rather than in hardware. The MDAC processing circuitry allows connectivity to be completely programmable. Basic circuit analysis techniques have been applied to demonstrate feasibility of implementing at least 300 neurons/chip with real-time response and about 1000 neurons/chip for non real-time response. As technology improvements are made, both of these figures will increase.

Functionality has been demonstrated with breadboard circuit, and the integrated circuit is currently being fabricated. Future work could involve extending the number of storage elements per chip, increasing the number of possible weighting factors, including an on-board digital microcontroller for enhanced circuit performance, providing a digital to analog converter to allow digital storage of threshold voltages, and networking multiple chips together.

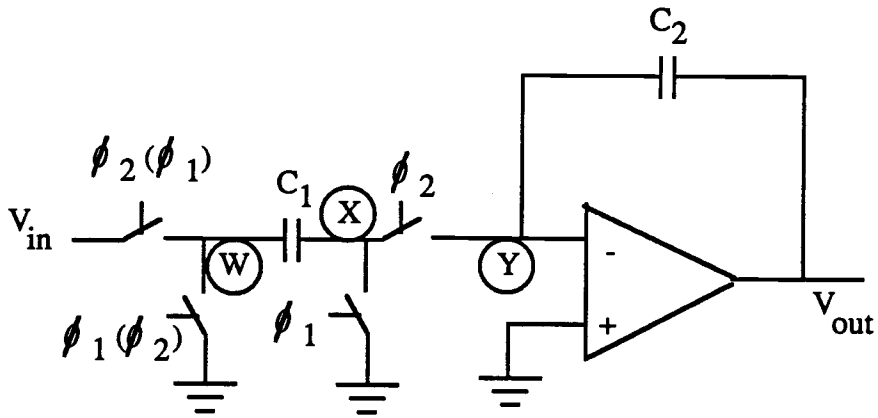


Figure 1. Switched-capacitor integrator

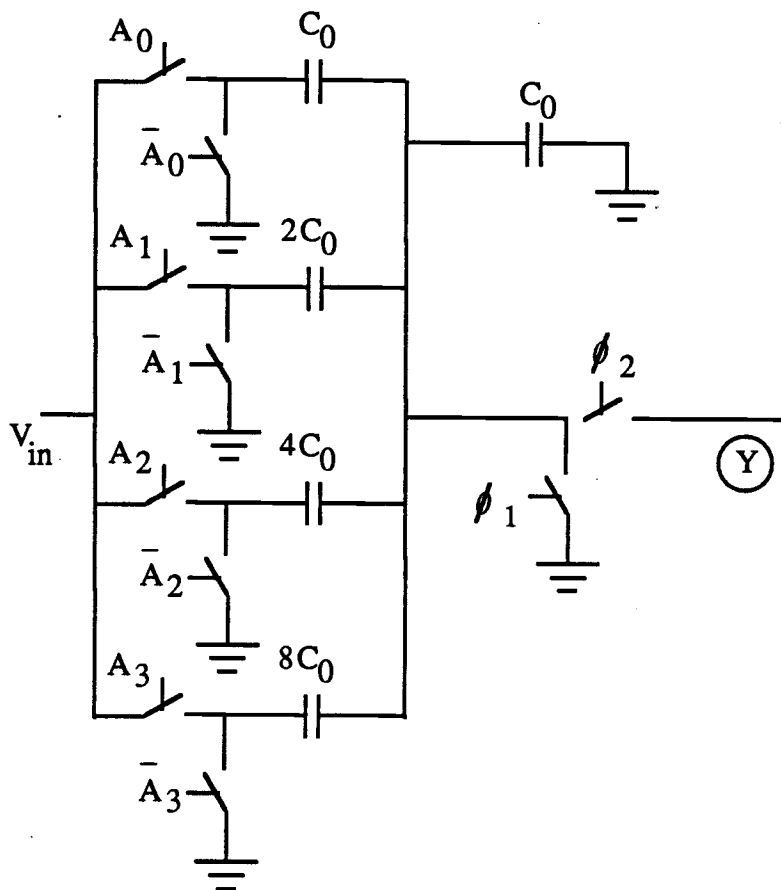


Figure 2. 4 bit MDAC

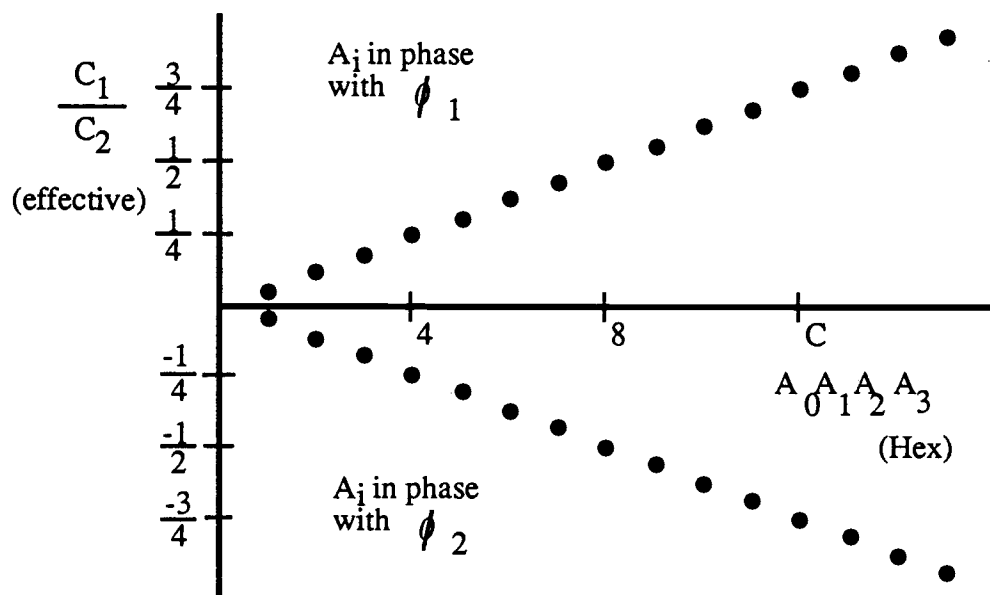


Figure 3. Possible weighting factors

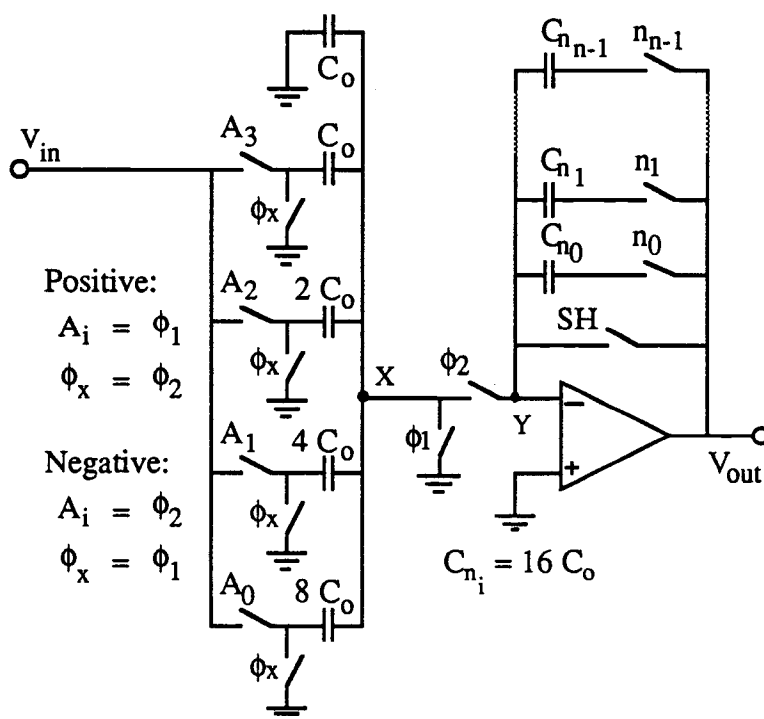
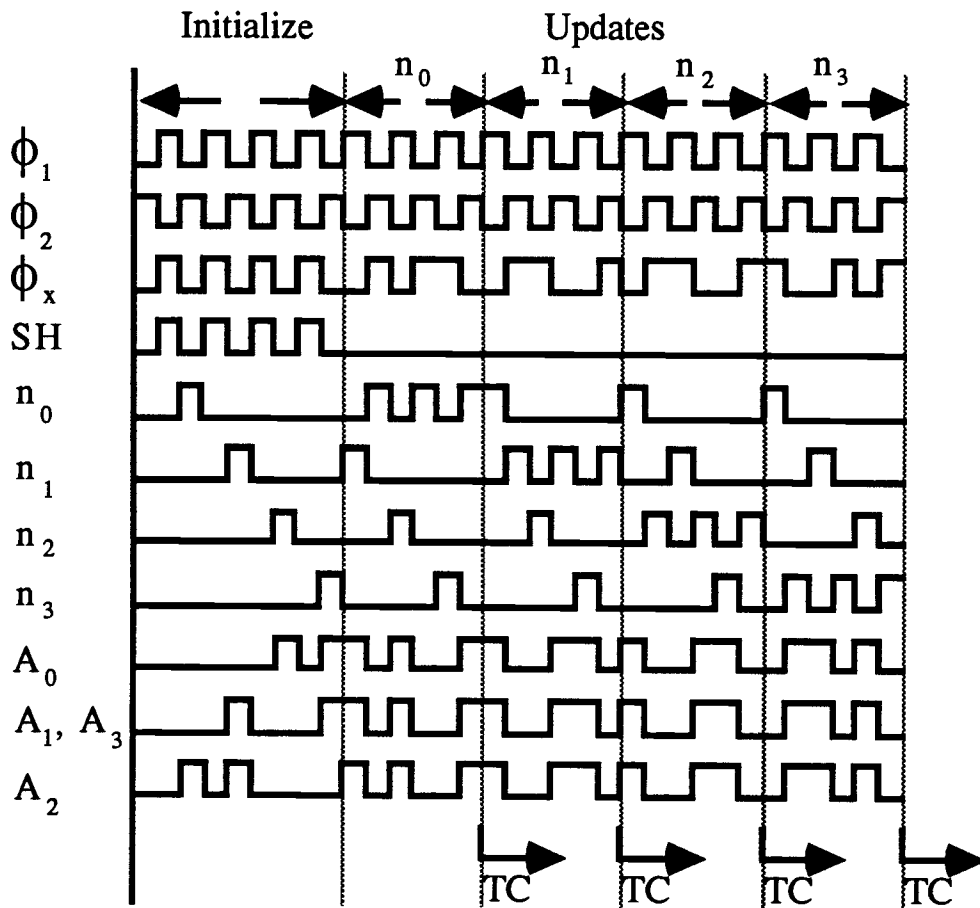


Figure 4. Programmable time-multiplexed integrator



TC = Threshold Compare

$$V_{n0} = -2/16 V_{in} \quad V_{n1} = -7/16 V_{in}$$

$$V_{n2} = -8/16 V_{in} \quad V_{n3} = -14/16 V_{in}$$

Figure 5. Example control signals

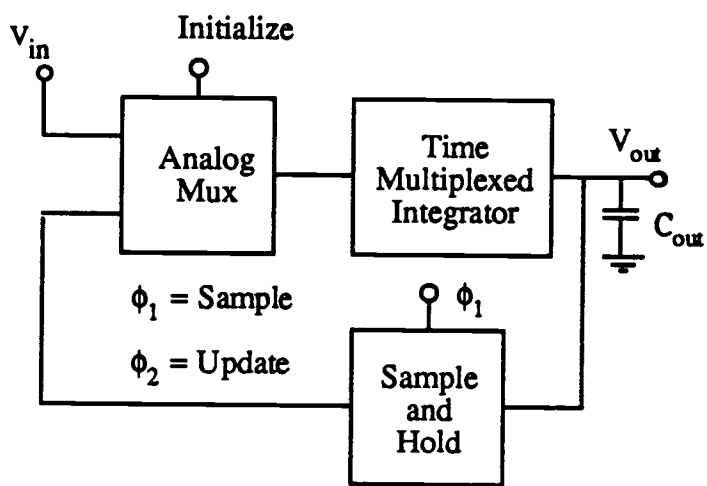


Figure 6. Sequential sample and update circuitry

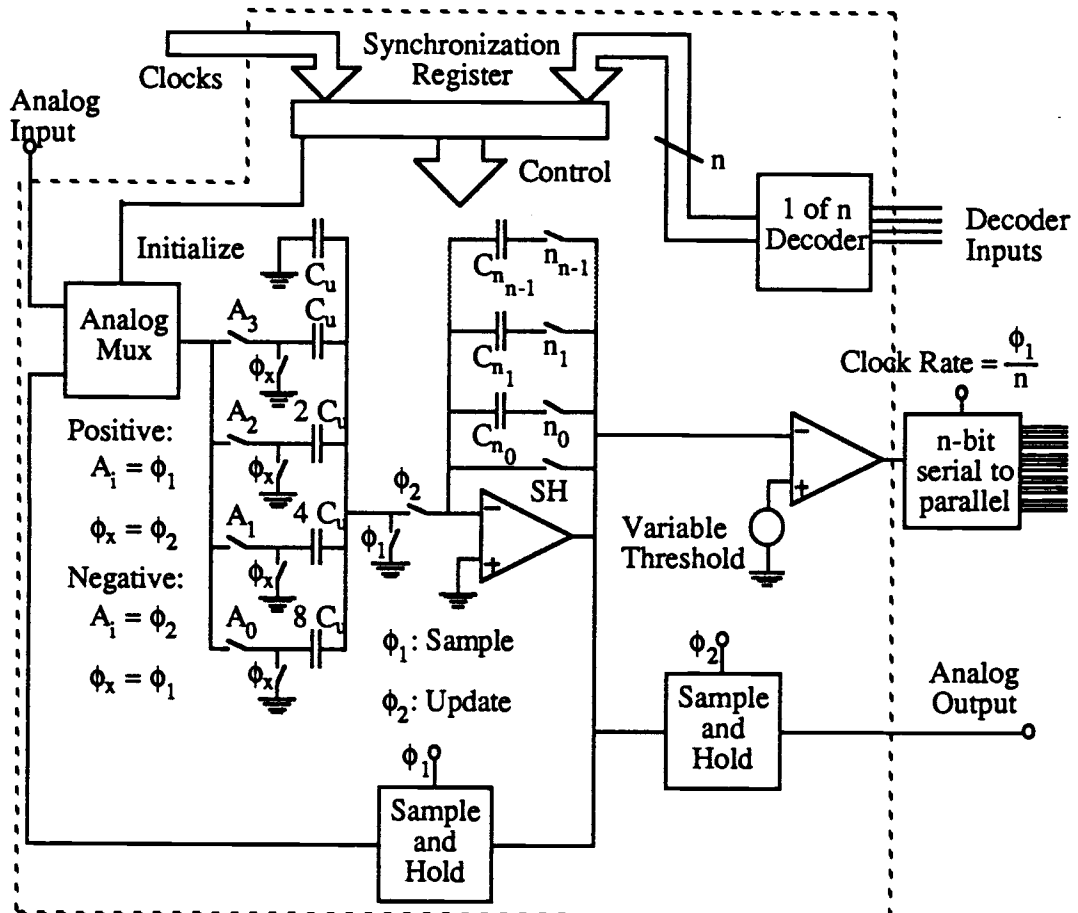


Figure 7. Complete circuit

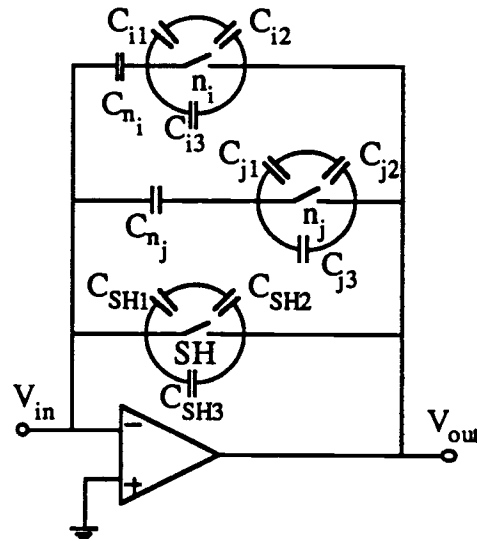


Figure 8. Parasitic capacitance model

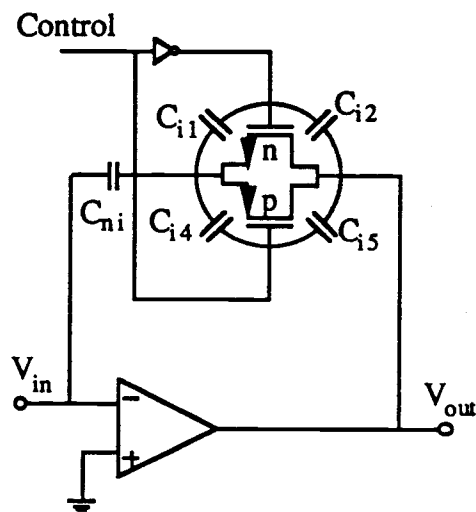


Figure 9. T-gate parasitic capacitance model

<u>Part No.</u>	<u>Description</u>	<u>Quantity</u>
LF356	Wide-band JFET Op-amp	4
LF13331	Quad JFET analog switch (± 12)	7
7402	Quad NOR	1
7404	Quad inverter	4
74138	3-8 decoder with enable	2
LM302	Wide-band unity gain buffer	1
LM339	Voltage comparator	1
74164	8 bit serial - parallel shift register	1
74198	8 bit parallel-in/parallel-out shift register	5
4066	Quad analog switch (+5, 0)	4
7474	Dual D flip-flop	1
-----	Light emitting diode (LED)	16
-----	500 Ω current limiting resistor	16
-----	10 K Ω pull-down resistor	16
-----	1000 pF mica capacitor for sample and hold	1
-----	270 pF mica capacitor for neurons	16
-----	270 pF mica capacitor for MDAC	1
-----	Diodes for rectifier	2
-----	4.5 K Ω resistors for rectifier	2
-----	6 K Ω resistors for V_T	2

Table 1. List of parts for breadboard circuit

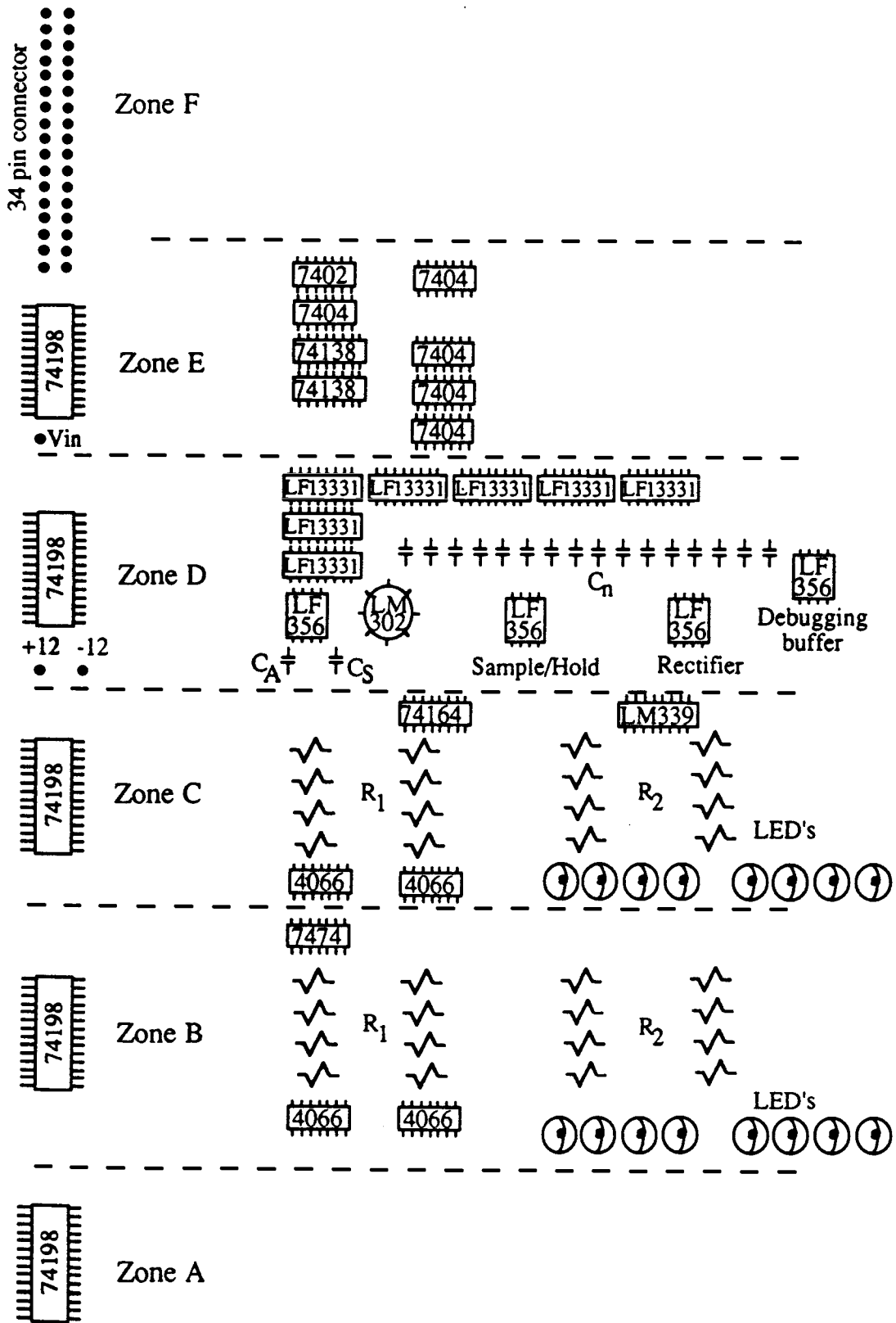


Figure 10. Breadboard circuit layout

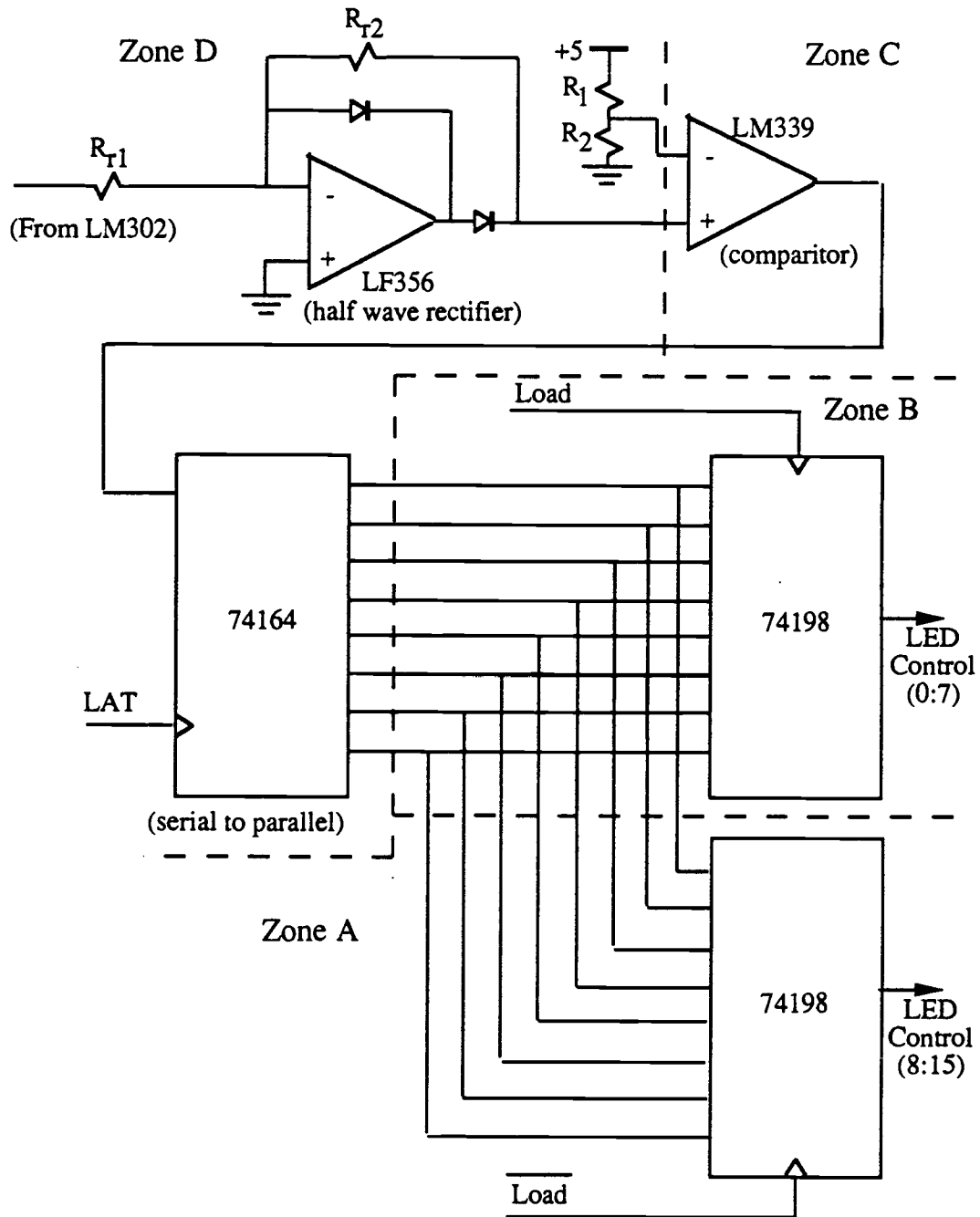


Figure 12. Half wave rectifier, comparator, and serial-parallel conversion

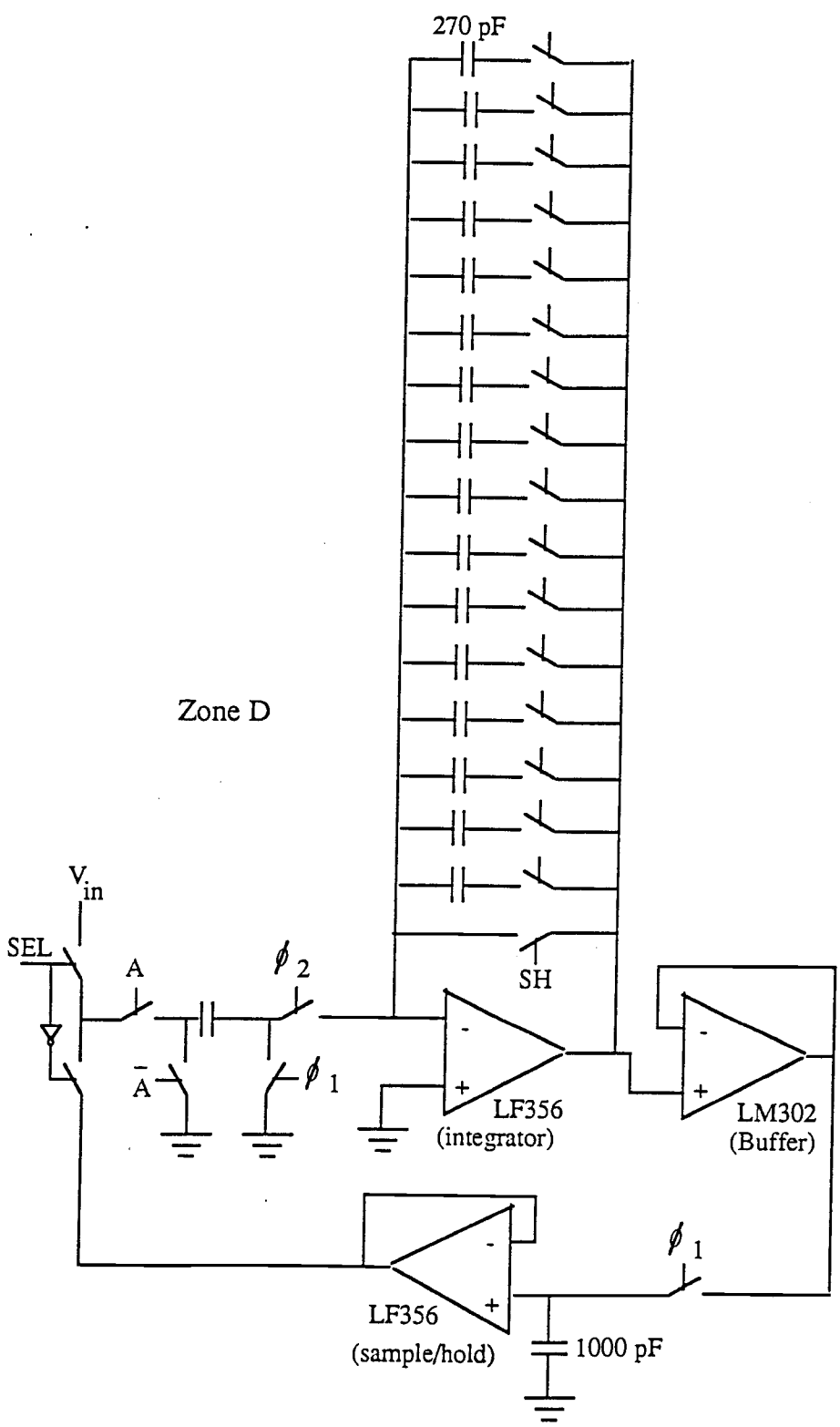


Figure 13. Integrator, buffer, and sample/hold

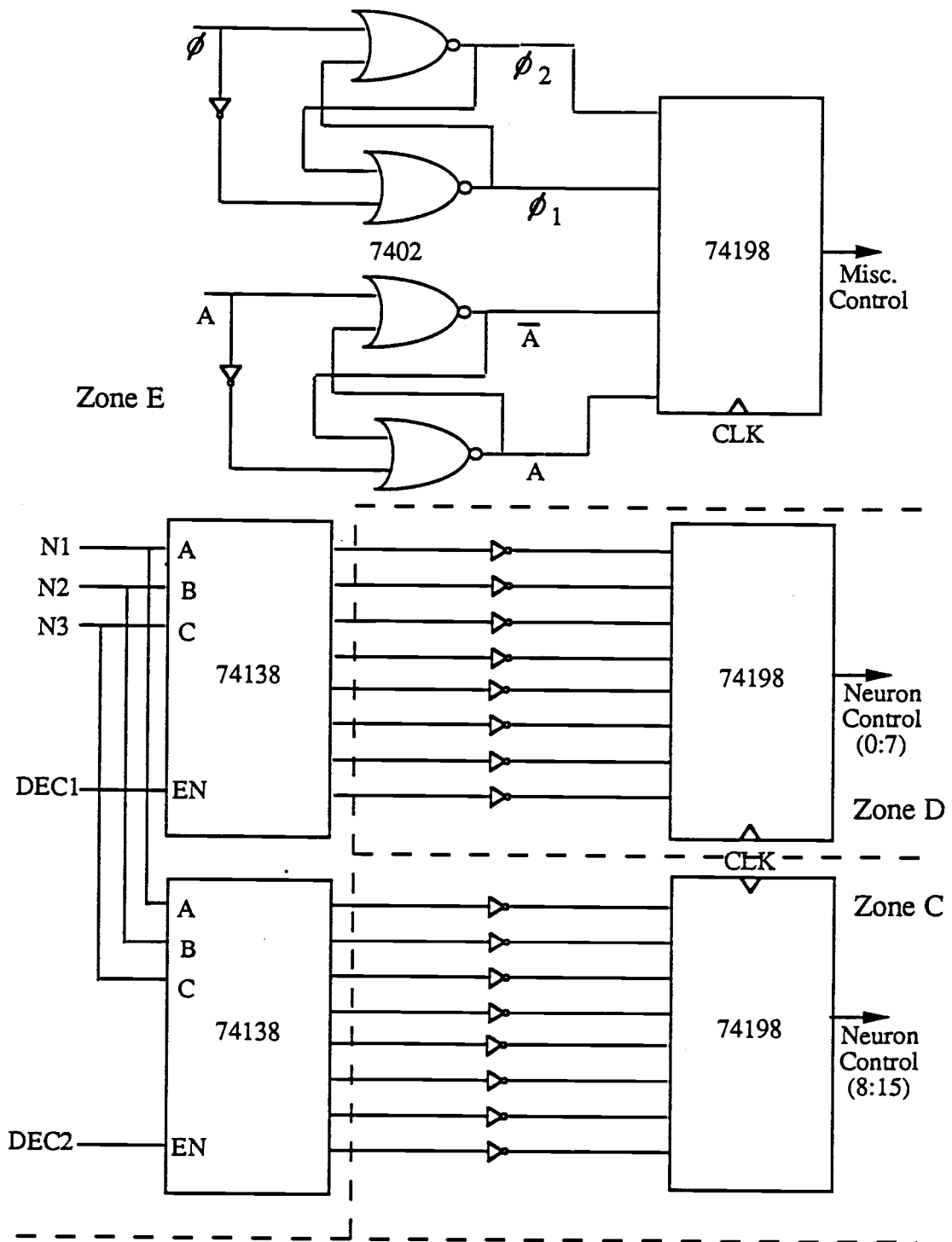


Figure 14. Digital Control Circuitry

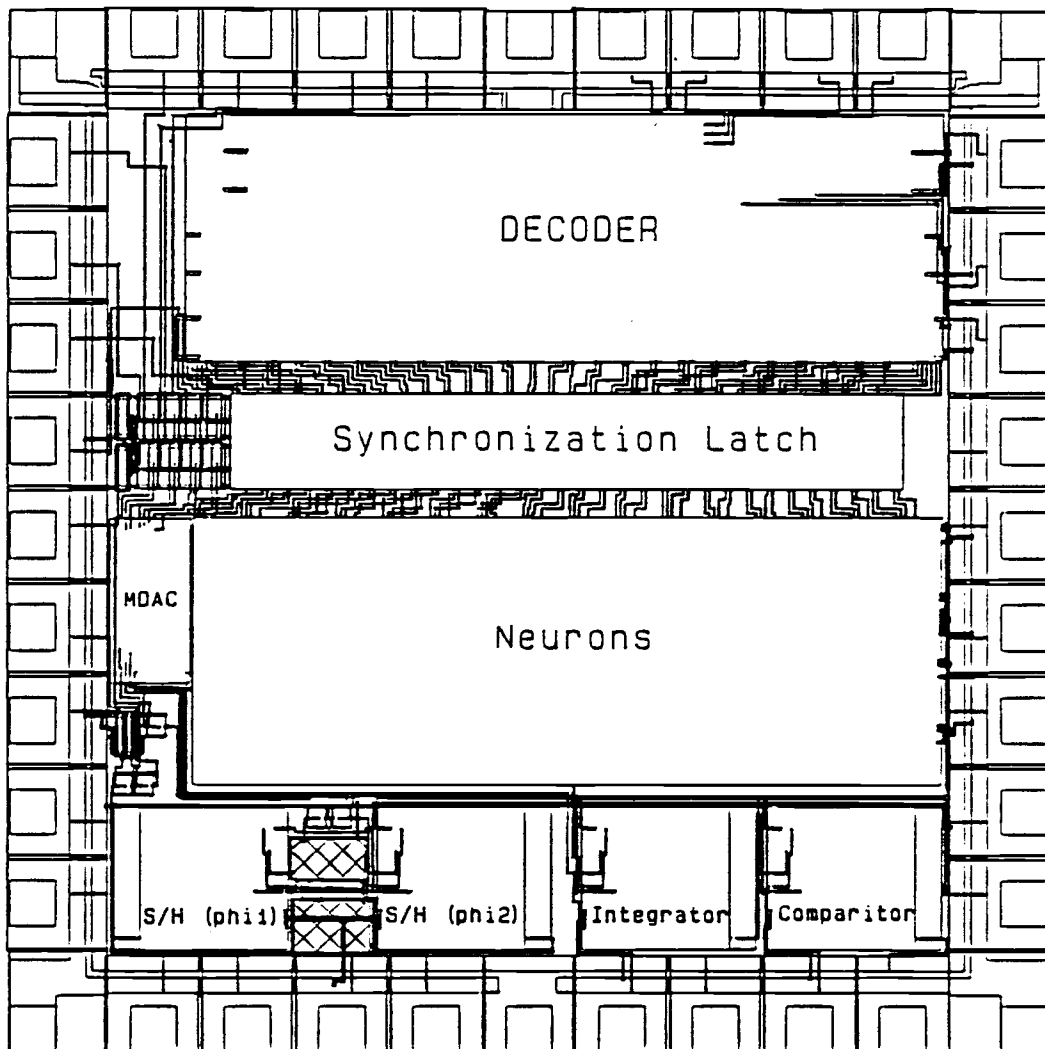


Figure 15. Integrated circuit block diagram

<u>Pin No.</u>	<u>Name</u>	<u>Pin No.</u>	<u>Name</u>
1	GND	21	VCC
2	UNUSED	22	VREF (VCC/2)
3	A1 (MDAC C_0)	23	UNUSED
4	A2 (MDAC $2C_0$)	24	V_T
5	A3 (MDAC $4C_0$)	25	DIGITALOUT
6	VCC	26	GND
7	A4 (MDAC $8C_0$)	27	VBIAS1
8	SH	28	VBIAS2
9	ϕ_x	29	UNUSED
10	ϕ	30	UNUSED
11	CLOCK	31	UNUSED
12	ANALOGIN (V_{IN})	32	N51 (TEST)
13	SEL	33	B (DEC_IN)
14	UNUSED	34	C (DEC_IN)
15	UNUSED	35	A (DEC_IN)
16	GND	36	VCC
17	UNUSED	37	F (DEC_IN)
18	UNUSED	38	G (DEC_IN)
19	UNUSED	39	D (DEC_IN)
20	ANALOGOUT	40	G (DEC_EN)

Table 2. Integrated circuit pinout

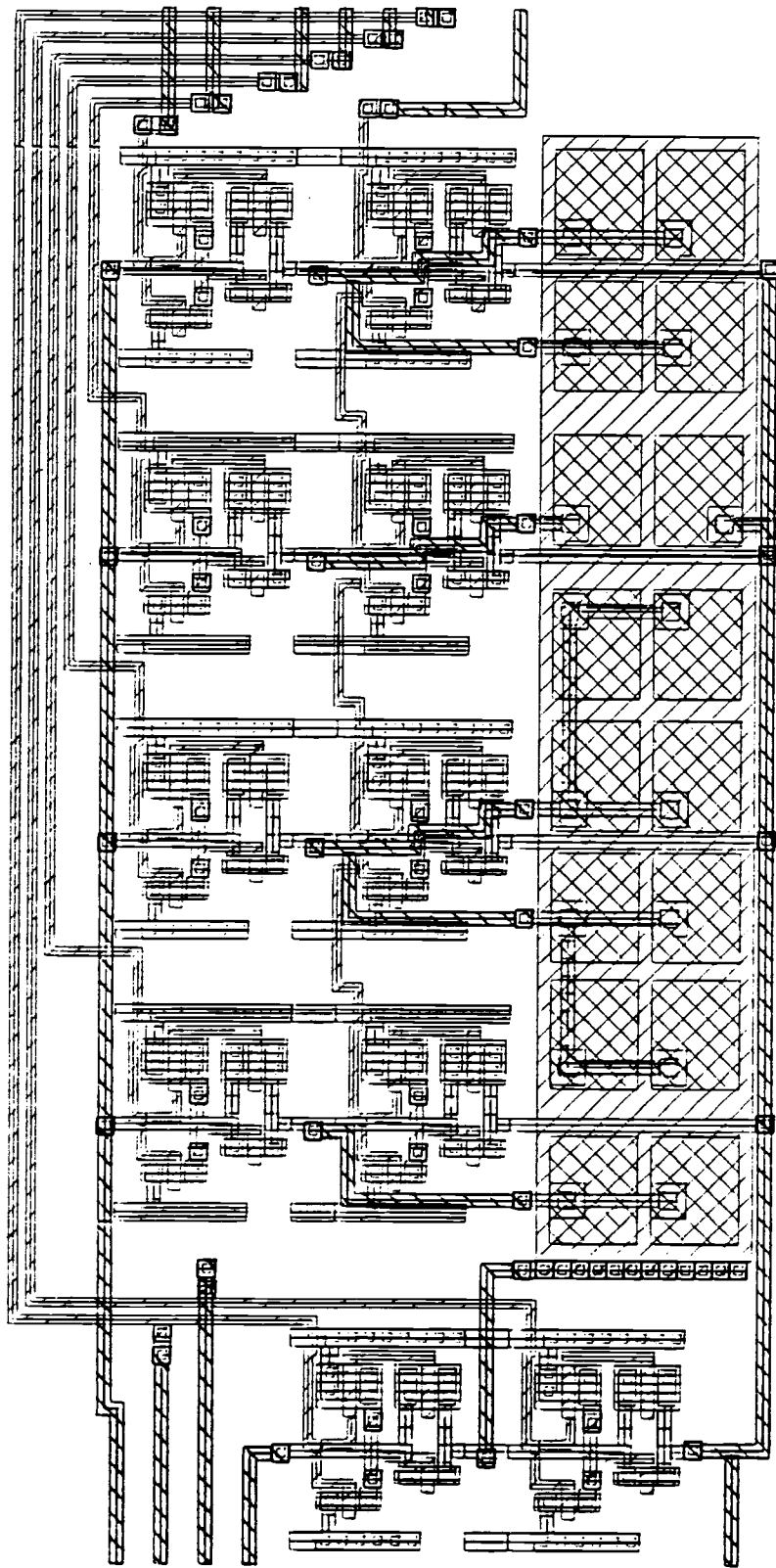


Figure 16. MDAC layout

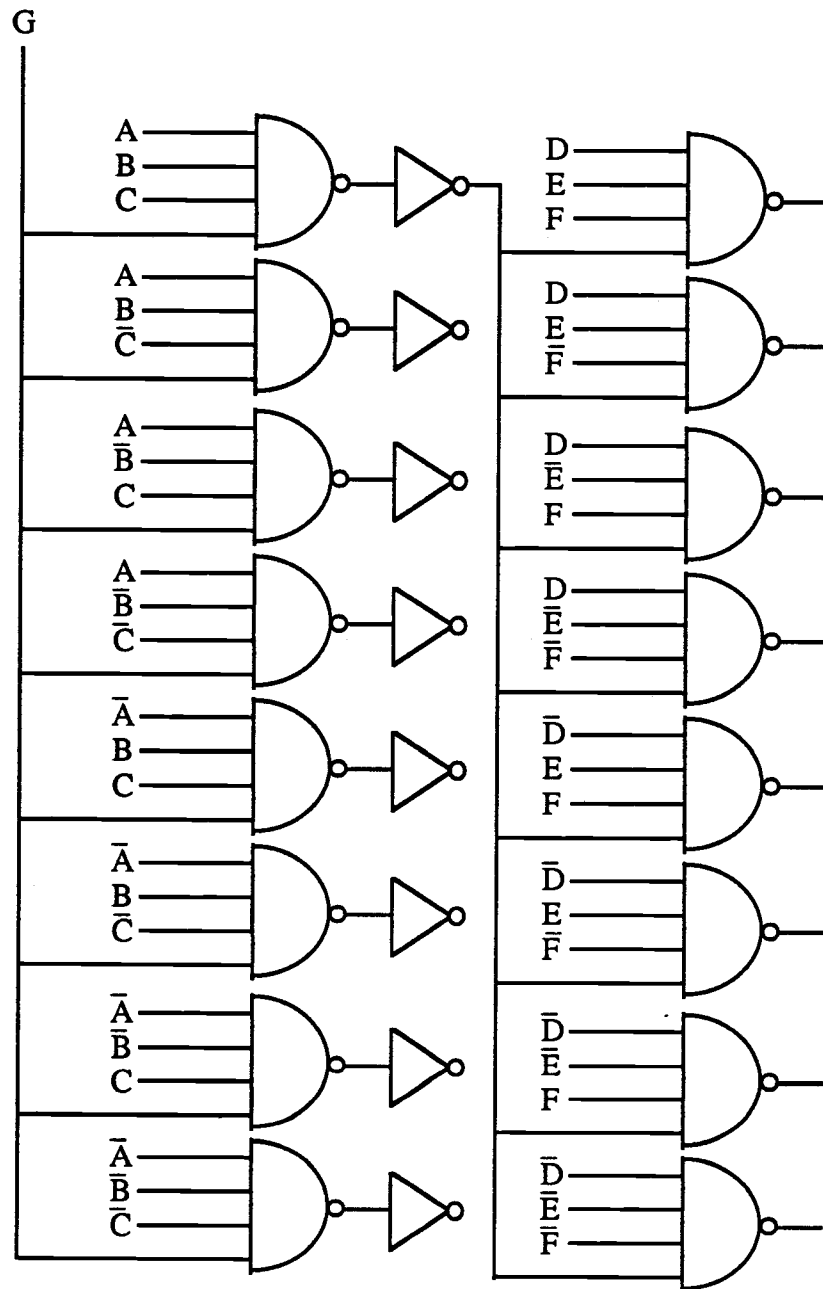


Figure 17. Decoder input stage and 8 output stages

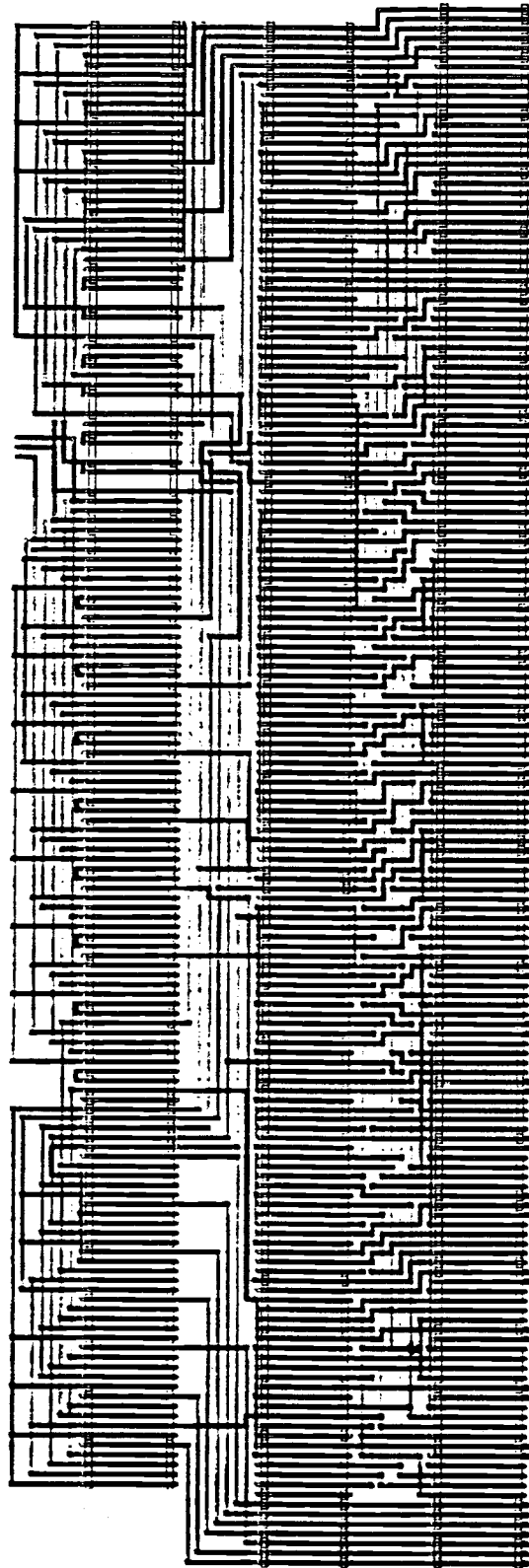


Figure 18. Static 6-64 decoder floorplan

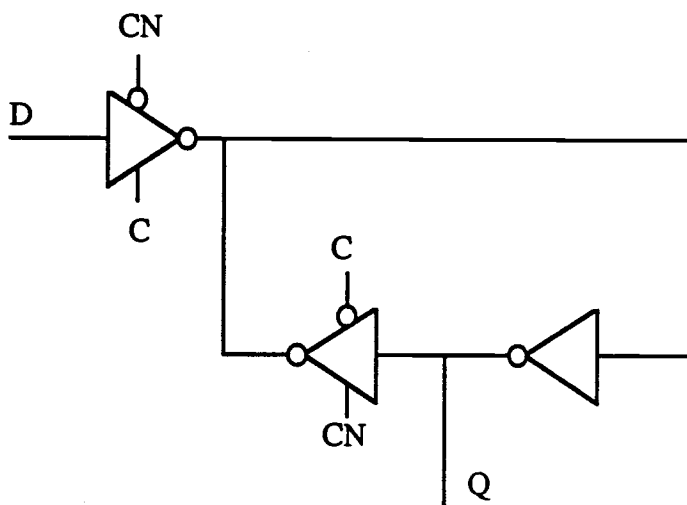


Figure 19. Synchronization latch

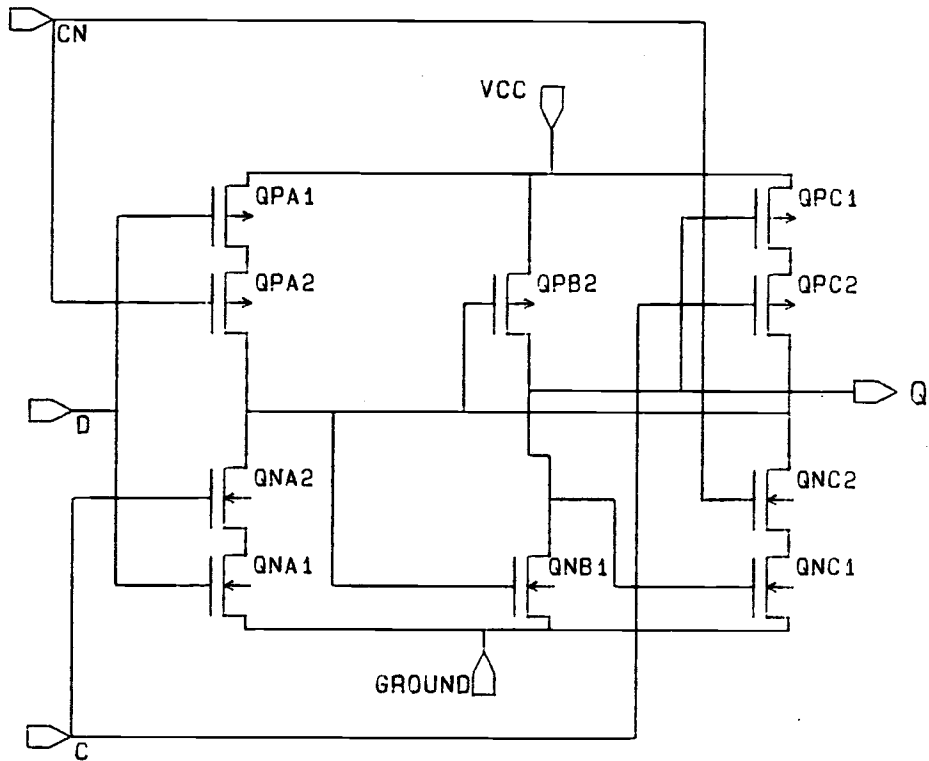


Figure 20. Latch transistor diagram

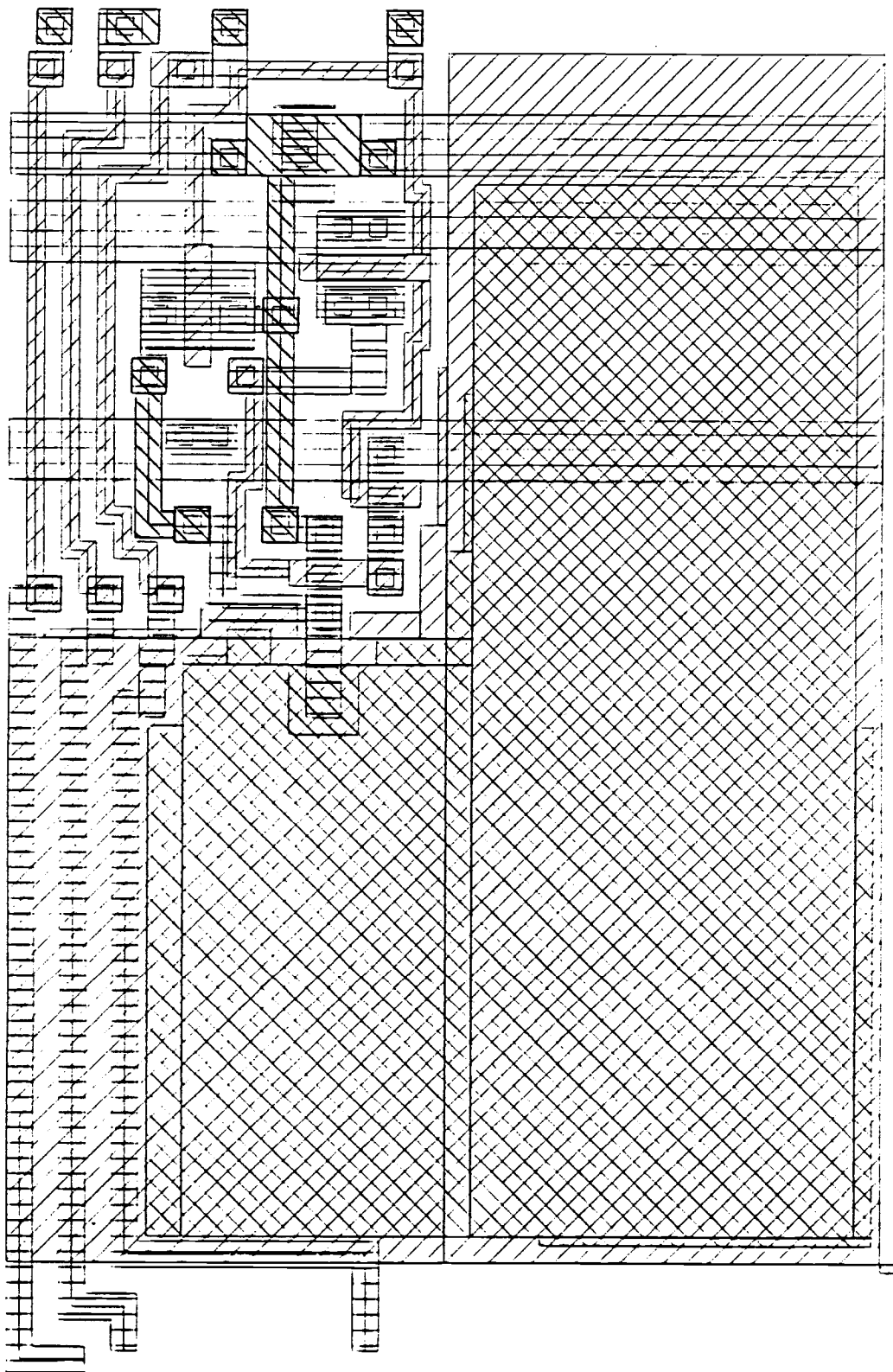


Figure 22. Layout of neuron access circuitry

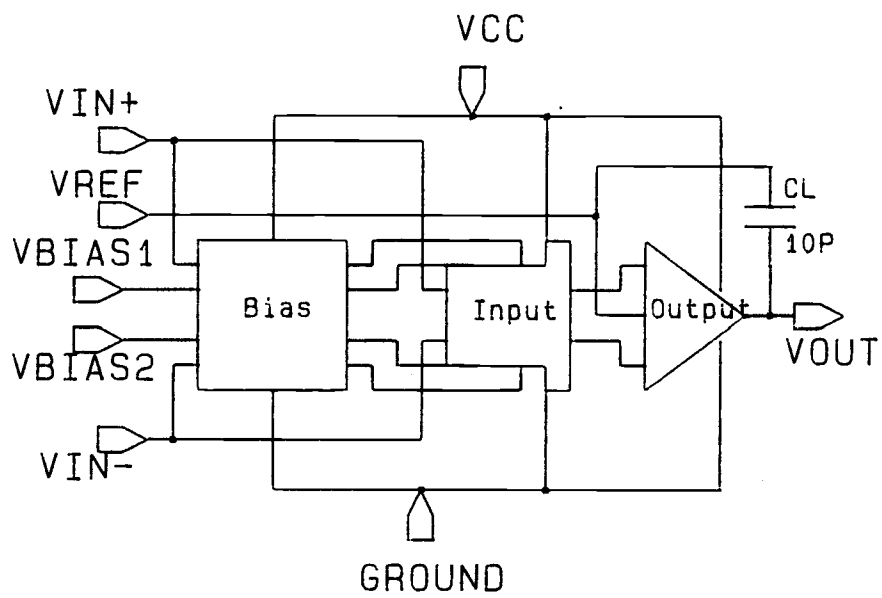
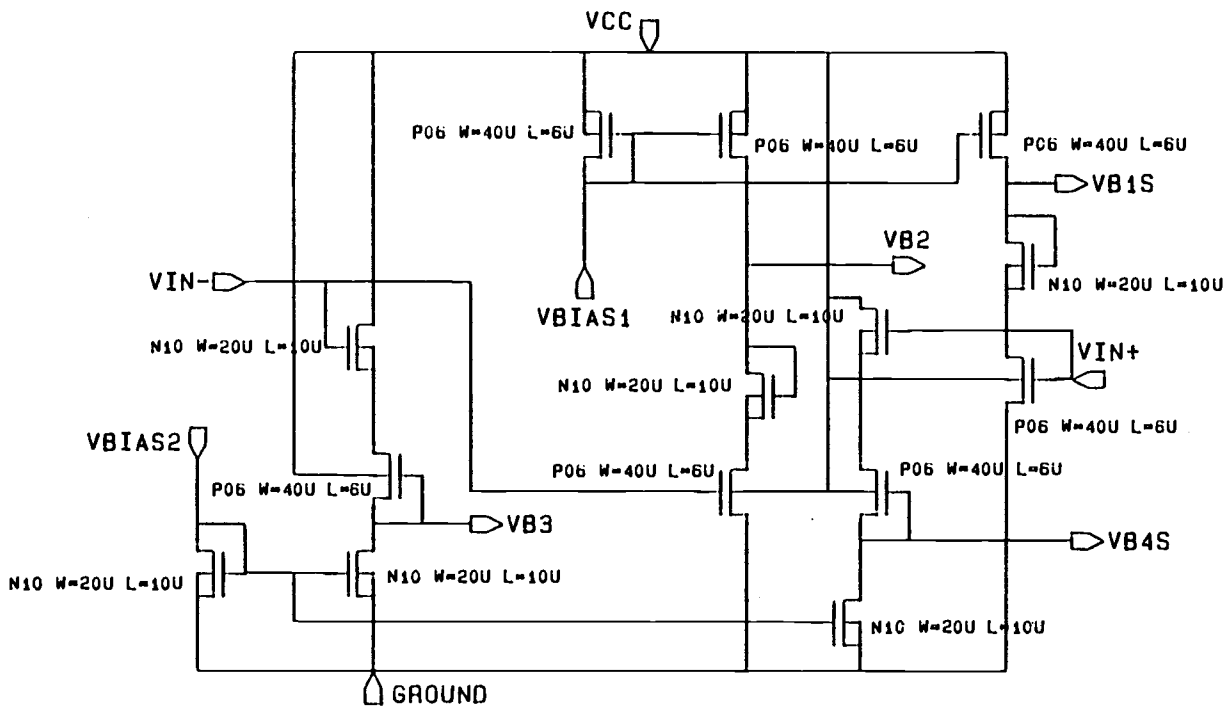


Figure 23 (a). Op-amp block diagram

Figure 23 (b). Op-amp biasing circuitry



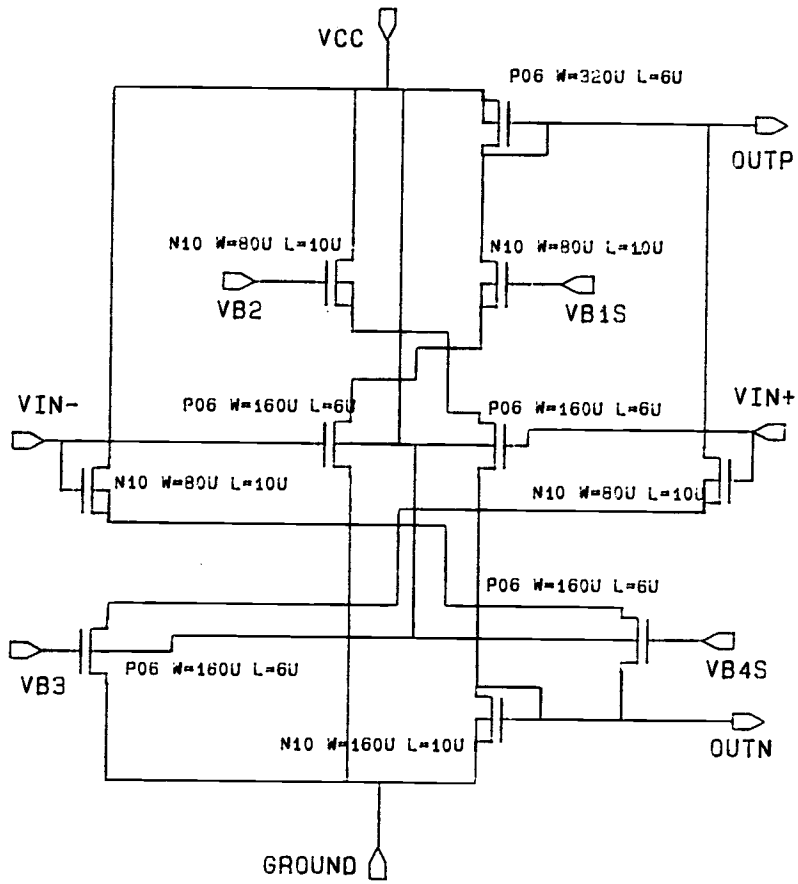


Figure 23(c). Op-amp input stage

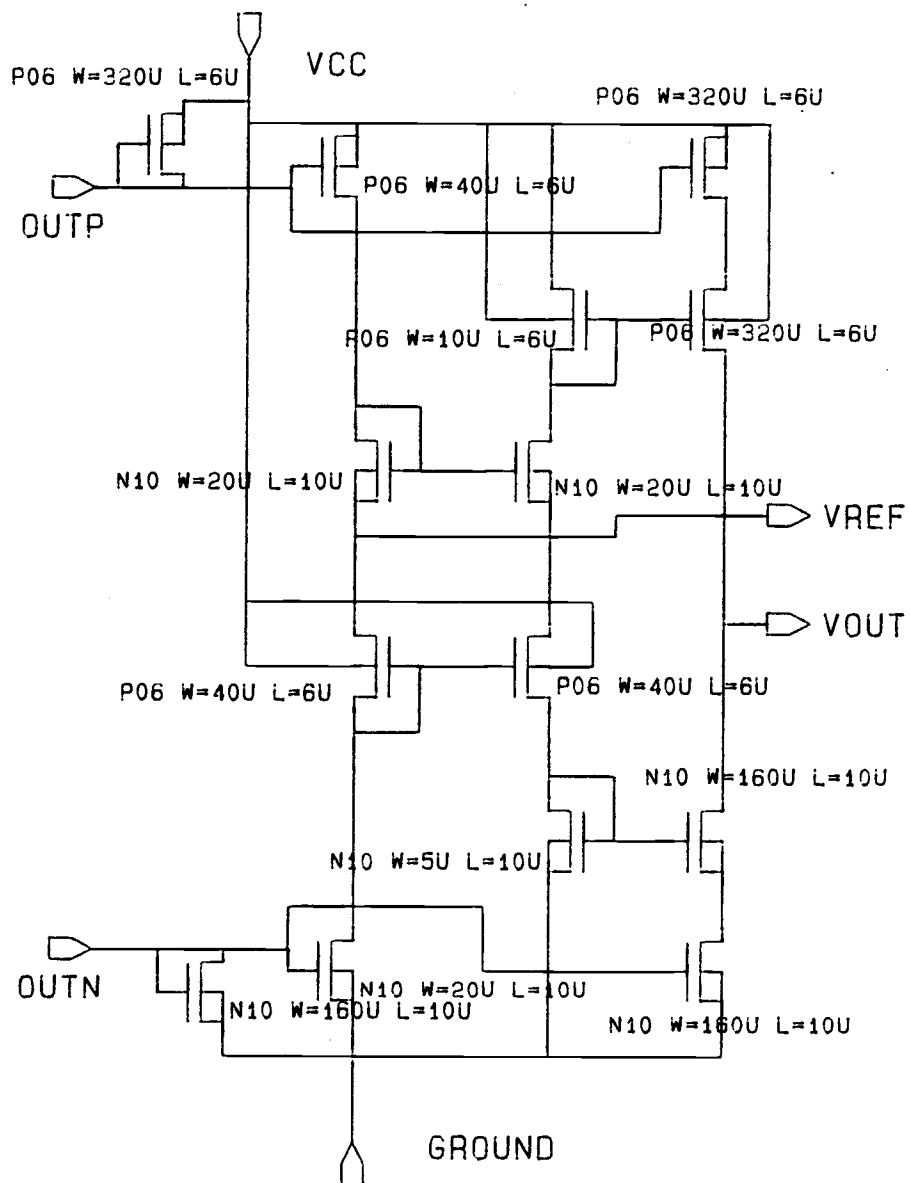


Figure 23(d). Op-amp output stage

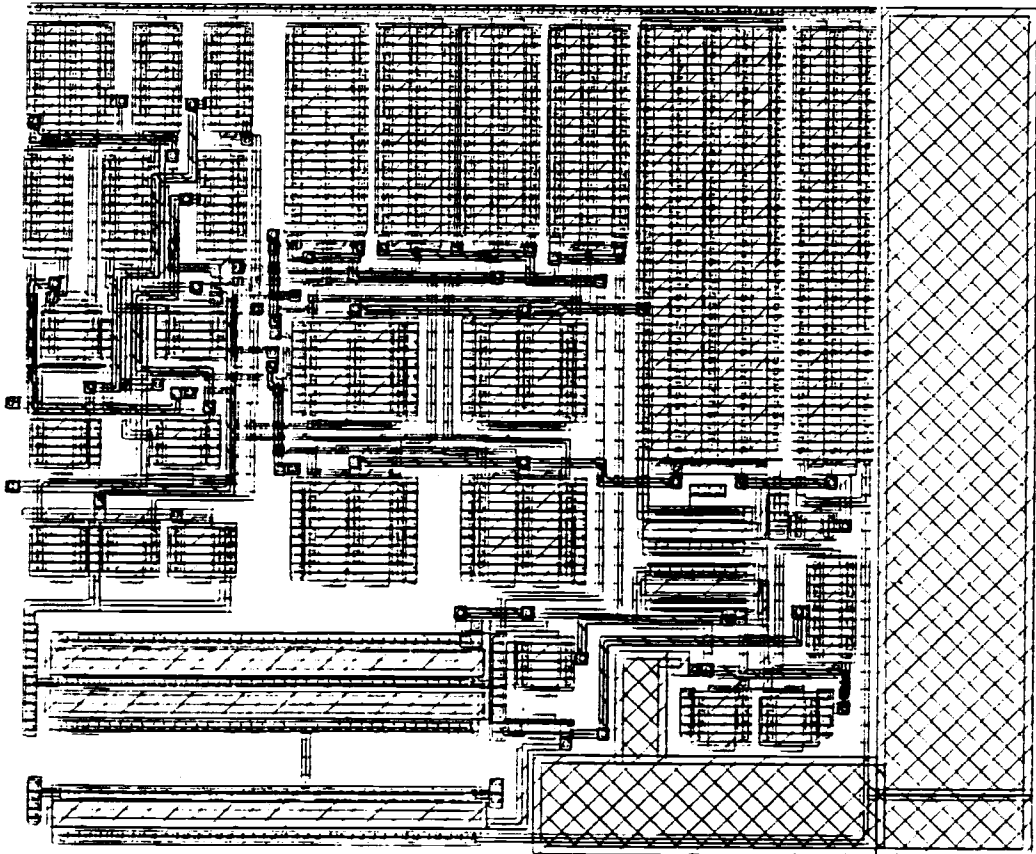


Figure 24. Op-amp layout

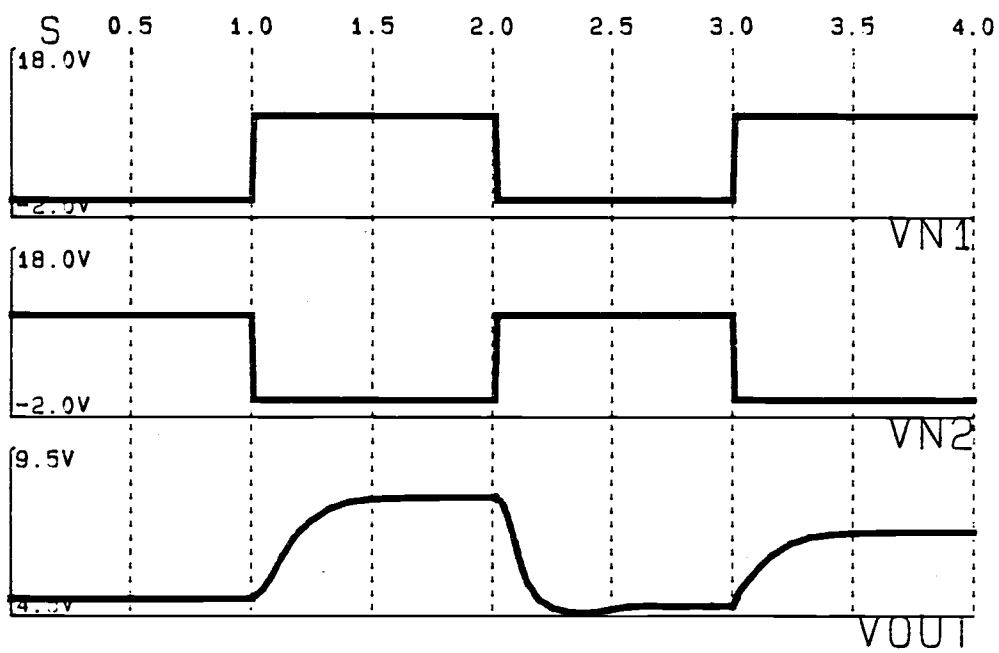


Figure 25.

Integrator output relative to neuron access (SPICE)

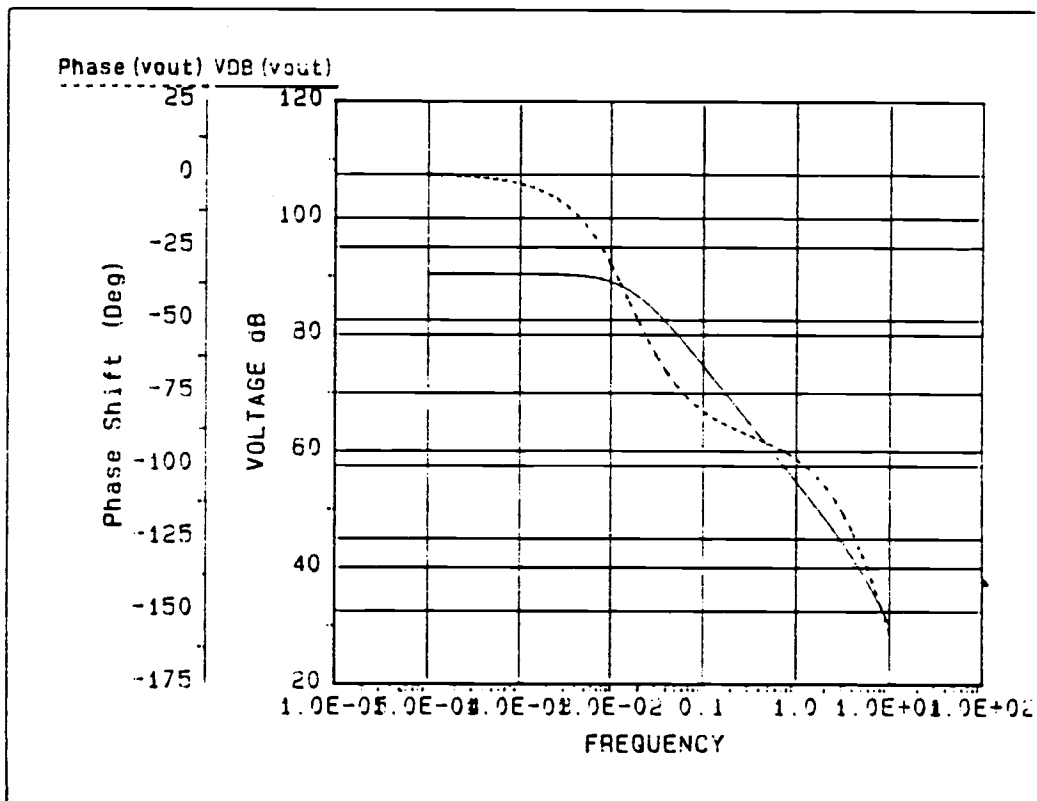


Figure 26. Op-amp open loop frequency response

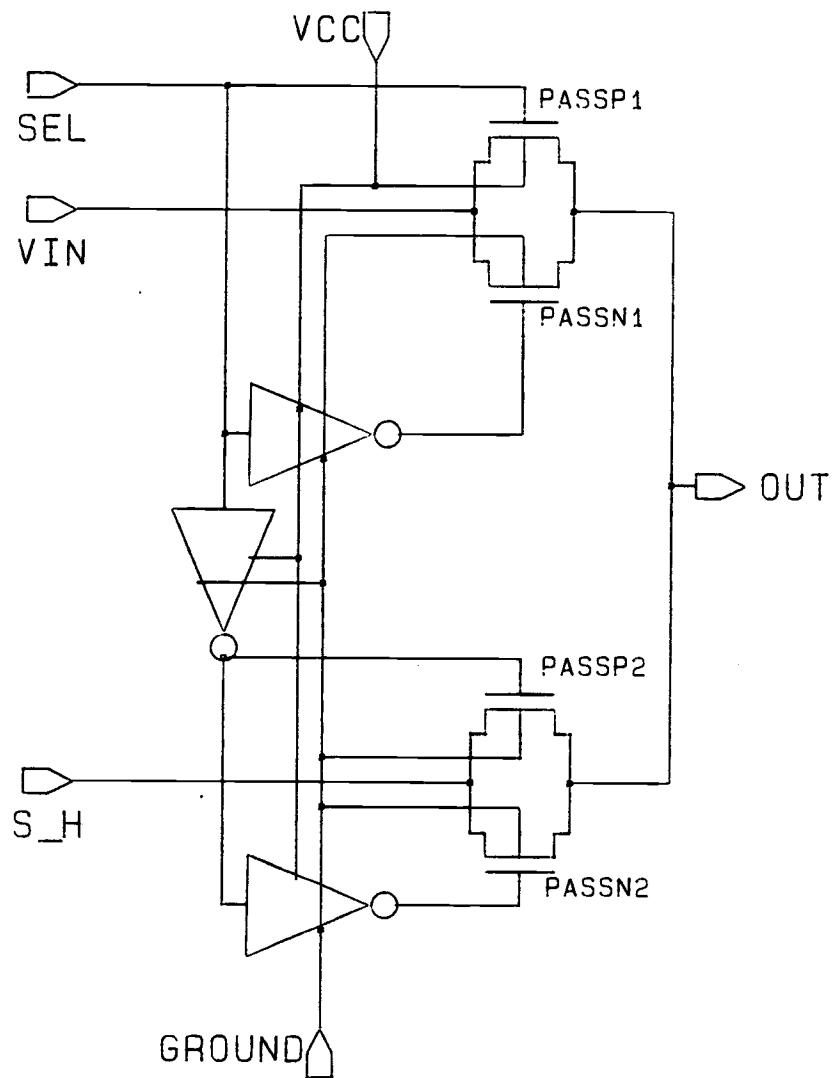
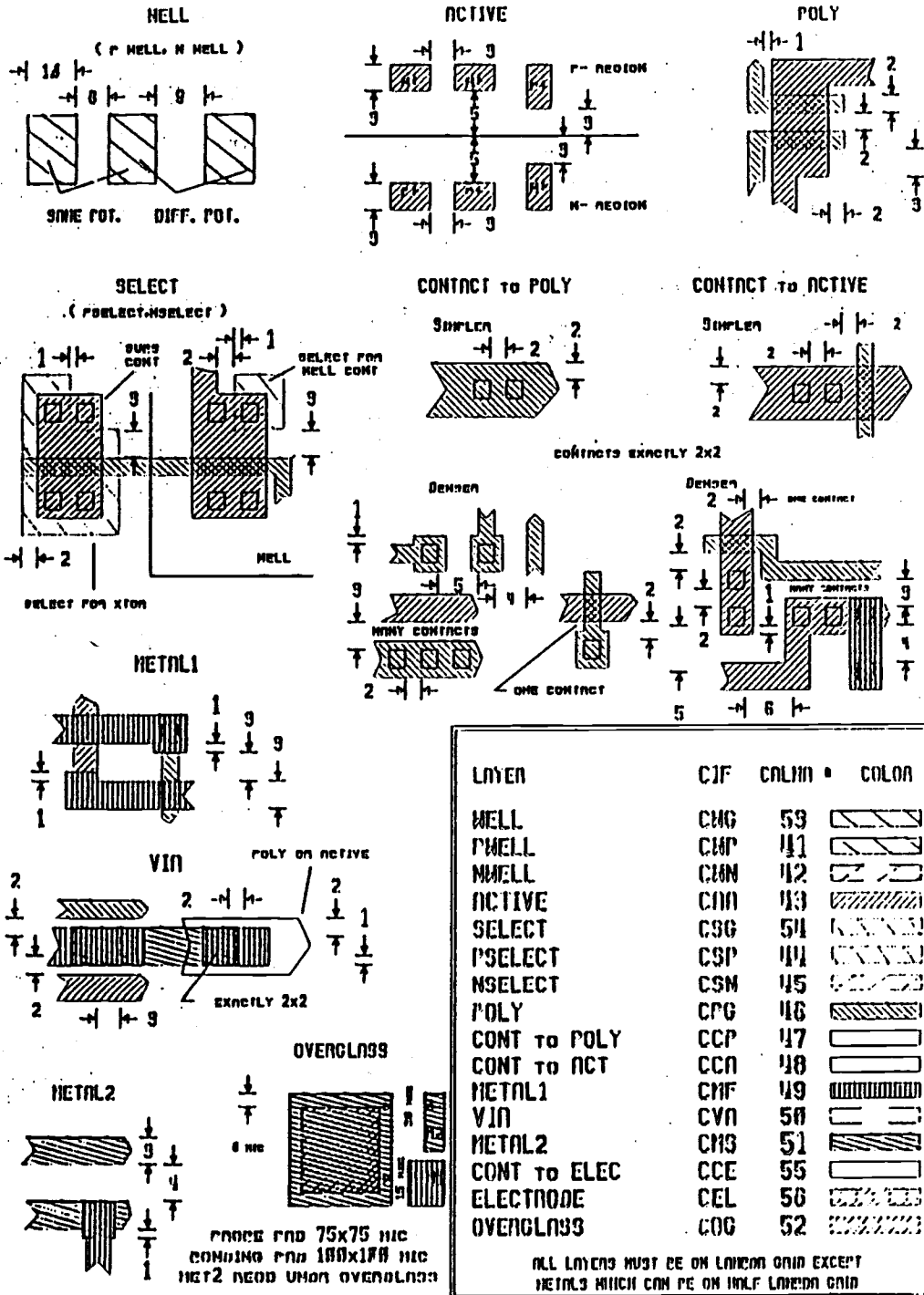


Figure 27. Analog multiplexor

NOSIS CMOS SCALABLE RULES (REV G)



LAYER	CIF	CALLIN	COLOR
WELL	CMG	59	[Pattern]
PWELL	CMF	41	[Pattern]
NWELL	CMN	42	[Pattern]
ACTIVE	CAN	43	[Pattern]
SELECT	CSG	54	[Pattern]
PSELECT	CSP	44	[Pattern]
NSELECT	CSN	45	[Pattern]
POLY	CPG	46	[Pattern]
CONT TO POLY	CCP	47	[Pattern]
CONT TO ACT	CCN	48	[Pattern]
METAL1	CMF	49	[Pattern]
VIN	CVN	50	[Pattern]
METAL2	CMG	51	[Pattern]
CONT TO ELEC	CCE	55	[Pattern]
ELECTRODE	CEL	56	[Pattern]
OVERGLASS	COG	52	[Pattern]

ALL LAYERS MUST BE ON LAYER 0 AND EXCEPT METALS WHICH CAN BE ON HALF LAYER 0 AND

Figure 28. DRC rules

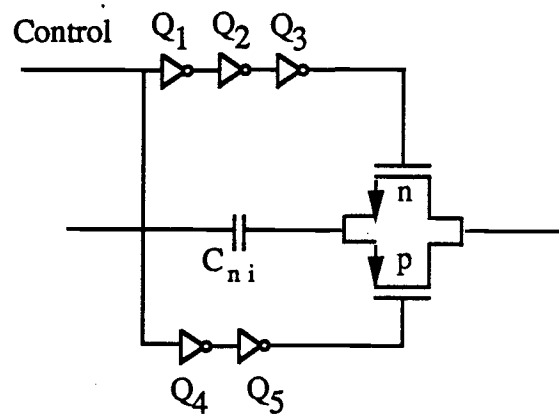


Figure 29. Improved neuron access circuitry

VIII. REFERENCES

- [1] D.E. Rumelhart, J.L. McClelland, and the PDP Research Group: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition Volume 1: Foundations*. [MIT Press, 1986], p. 130.
- [2] J.J. Hopfield and D.W. Tank: "Computing with Neural Circuits: A Model," *Science* v. 233, n. 4764 [August 8, 1986], p.131.
- [3] H.P. Graf, L.D. Jackel, R.E. Howard, B. Straughn, J.S. Denker, W. Hubbard, D.M. Tennant, and D. Schwartz: "VLSI Implementation of a Neural Network Memory with Several Hundreds of Neurons," *Neural Networks for Computing AIP Conference Proceedings* n. 151 [1986], pp. 182-187.
- [4] J. Bailey and D. Hammerstrom: "Why VLSI Implementations of Associative VLCNs Require Connection Multiplexing," *Proceedings of the 1988 IEEE International Conference on Neural Networks* v. 2, pp. 173-180.
- [5] J.G. Cleary: "A Simple VLSI Connectionist Architecture," *IEEE First Annual Neural Network Conference*, v. 3 [1987], pp. 419-426.
- [6] R. Gregorian and G. Temes: *Analog MOS Integrated Circuits for Signal Processing*, [Wiley - Interscience, 1986], p. 277-279.
- [7] Y.P. Tsvividis and D. Anatassiou: "Switched-Capacitor Neural Networks," *Electronics Letters*, v. 23 [August, 1987], pp. 958-959.
- [8] A. Rodríguez-Vázquez, A. Rueda, J.L. Huertas, and R. Domínguez-Castro: "Switched-Capacitor Neural Networks for Linear Programming," *Electronics Letters*, v. 24 [April, 1988], pp. 496-497.
- [9] "The Silicon Association Cortex", seminar presentation by D. Hammerstrom, Oregon State University, June 2, 1989.
- [10] *Linear Databook*, National Semiconductor Corporation, pp.3-22 - 3-41, 6-17 - 6-26, 1982.

- [11] "Analysis and Synthesis of Neural Networks", MS thesis of Jeanette Skelton, Oregon State University, 1989.
- [12] "MOSIS Scalable CMOS Design Rules & Process Parameters", University of Southern California, 1988.
- [13] "A High-Swing Class-AB CMOS Operational Amplifier", MS thesis of Choung Yu, Oregon State University, 1988.
- [14] J.J. Hopfield: "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Science, USA*, v. 79 [April, 1982], pp. 2554-2558.

IX. APPENDICES

APPENDIX 1. Parasitic capacitance equation derivation

For the switch closed case, the model for parasitic capacitance can be reduced to the circuit of figure A1.

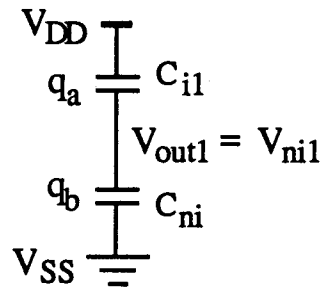


Figure A1. Switch closed

This circuit implies the following equations,

$$q_a = -(V_{DD} - V_{ni1}) C_{i1}$$

$$q_b = V_{ni1} C_{ni}$$

When the switch closes, the equivalent circuit becomes that of figure A2.

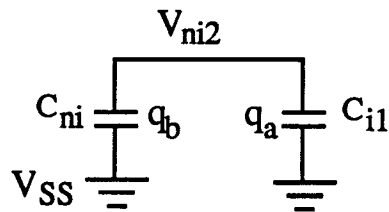


Figure A2. Switch open

From conservation of charge, the total charge remains the same.

$$q_c = q_a + q_b = V_{n2} (C_{ni} + C_{i1})$$

Substituting,

$$\begin{aligned}
V_{ni2} (C_{ni} + C_{i1}) &= -(V_{DD} - V_{ni1}) C_{i1} + V_{ni1} C_{ni} \\
V_{ni2} &= -(V_{DD} - V_{ni1}) C_{i1} / (C_{i1} + C_{ni}) + V_{ni1} C_{ni} / (C_{i1} + C_{ni}) \\
\Delta V_{nc} = V_{ni2} - V_{ni1} &= -C_{i1} / (C_{i1} + C_{ni}) V_{DD}
\end{aligned} \tag{4}$$

A similar procedure is followed in analyzing the fully complementary T-gate of figure C. The model of figure A3 is used with the switch closed,

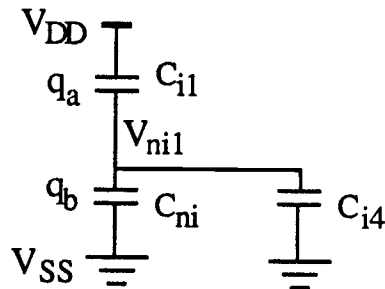


Figure A3. Switch closed

and with the switch open, the circuit can be modeled with figure A4

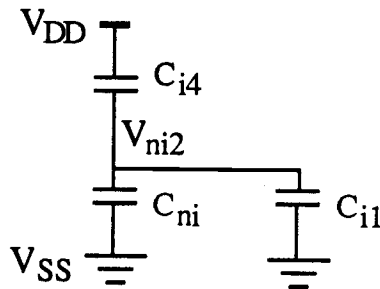


Figure A4. Switch open

implying the following equations,

$$\begin{aligned}
(V_{ni1} - V_{DD})C_{i1} + V_{ni1}(C_{i4} + C_{ni}) &= (V_{ni2} - V_{DD})C_{i4} + V_{ni2}(C_{i1} + C_{ni}) \\
V_{ni2}(C_{i1} + C_{i4} + C_{ni}) &= C_{i1} V_{ni1} - C_{i1} V_{DD} + C_{i4} V_{DD} + C_{ni} V_{ni1} + C_{i4} V_{ni1} \\
\Delta V_{nc} = V_{ni2} - V_{ni1} &= V_{DD} (C_{i4} - C_{i1}) / (C_{i1} + C_{i4} + C_{ni})
\end{aligned} \tag{5}$$

APPENDIX 2. SPICE parameters

```
.MODEL N10 NMOS LEVEL=2 LD=0.202970U TOX=394.000026E-10
+ NSUB=1.917E+16 VTO=0.971 KP=5.231000E-05 GAMMA=0.914
+ PHI=0.6 UO=598 UEXP=0.148 UCRIT=77508.0
+ DELTA=1.89 VMAX=75124 XJ=0.250000U LAMBDA=0.262E-02
+ NFS=1.749E+12 NEFF=1 NSS=1.000000E+12 TPG=1.000000
+ RSH=22.0001 CGDO=1.77E-10 CGSO=1.77E-10 CGBO=1.786E-10
+ CJ=4.10600E-04 MJ=0.44400 CJSW=3.560000E-10 MJSW=0.23800 PB=0.76000
* Weff = Wdrawn - Delta_W
* The suggested Delta_W is 0.27 um
.MODEL P06 PMOS LEVEL=2 LD=0.116U TOX=394.000026E-10
+ NSUB=8.33E+15 VTO=-0.81 KP=2.070000E-05 GAMMA=0.602
+ PHI=0.6 UO=237 UEXP=0.2472 UCRIT=62573.0
+ DELTA=0.102528 VMAX=47030 XJ=0.250000U LAMBDA=1.81230E-02
+ NFS=.908E+12 NEFF=1.001 NSS=1.000000E+12 TPG=-1.000000
+ RSH=73.30002 CGDO=10.32E-11 CGSO=10.32E-11 CGBO=3.4461E-10
+ CJ=2.5200E-04 MJ=0.39300 CJSW=1.92400E-10 MJSW=-0.054600 PB=0.49000
* Weff = Wdrawn - Delta_W
* The suggested Delta_W is 0.24 um
```

APPENDIX 3. DRC command file

```

; DRC Command File for double Metal/double Poly CMOS Process
;
;TRANSLATE-PROCESS = //kevily/users/hansen1/des/pdf.bin
;TRANSLATE-INPUT   = //kevily/users/hansen1/des/ho
;TRANSLATE-GO      = yes
*DESCRIPTION
PRIMARY   = HO
INDISK    = \\INGDS2
OUTDISK   = DRCOUT
PRINTFILE = DRCPRT
MODE      = EXEC NO
SYSTEM    = GDS2
SCALE     = 0.01 MICRON
RESOLUTION = 0.25 MICRON
LISTERROR = YES
KEEPDATA  = SMART
*END
*INPUT-LAYER
CWP = 8
CAA = 14
CPG = 15
CSP = 5
CCP = 18
CCA = 16
CMF = 12 TEXT 50 ATTACH = CMF
CVA = 11
CMS = 10
COG = 9
CEL = 3
CCE = 4
XP  = 52
CONNECT-LAY = CWP PSRCDRN NSRCDRN CAA CPG CMF CMS CEL XP
*END
*OPERATION
AND CAA CSP PDIF ;DEFINE ACTIVE P REGION
NOT CAA CSP NDIF ;DEFINE ACTIVE N REGION
AND CPG NDIF NGATE ;DEFINE N GATES
AND CPG PDIF PGATE ;DEFINE P GATES
OR NGATE PGATE GATE ;GENERIC GATES
SIZE GATE BY 1 EXGATE ;DEFINE
NOT CPG EXGATE FIELDP ;FIELD POLY
NOT NDIF NGATE NSRCDRN ;DEFINE N SOURCE/DRAINS
NOT PDIF PGATE PSRCDRN ;DEFINE P SOURCE/DRAINS
NOT CAA GATE SRCDRN ;GENERIC SOURCE/DRAINS
AND PSRCDRN CWP PSUB ;DEFINE PWELL CONTACT
AND PSUB CCA PCONT
NOT NDIF CWP NSUB ;DEFINE BULK CONTACT
AND NSUB CCA NCONT

```

```

CONNECT CME CPG BY CCP
CONNECT CME PSRCDRN BY CCA
CONNECT CME NSRCDRN BY CCA
CONNECT CME CAA BY CCA
CONNECT PSRCDRN CWP BY PCONT
CONNECT CEL CME BY CCE
CONNECT CME CMS BY CVA
CONNECT CMS XP BY COG

ELEMENT MOS [P] PGATE CPG PSRCDRN ;DEFINE P TRANSISTORS
ELEMENT MOS [N] NGATE CPG NSRCDRN CWP ;DEFINE N TRANSISTORS

;*****
;***** LAYER 1 CHECKING (PWELL) *****
;*****
WIDTH CWP LT 10.0 OUTPUT DRC11 30
EXT CWP LT 8.0 OUTPUT DRC12 30

;*****
;***** LAYER 2 CHECKING (ACTIVE) *****
;*****
WIDTH CAA LT 3.0 OUTPUT DRC21 30
EXT CAA LT 3.0 OUTPUT DRC22 30
ENC [T] NSRCDRN CWP LT 5.0 OUTPUT DRC23A 30
EXT PSRCDRN CWP LT 5.0 OUTPUT DRC23B 30
ENC [T] PSUB CWP LT 3.0 OUTPUT DRC24A 30
EXT NSUB CWP LT 3.0 OUTPUT DRC24B 30

;*****
;***** LAYER 3 CHECKING (POLY) *****
;*****
WIDTH CPG LT 2.0 OUTPUT DRC31 30
EXT CPG LT 2.0 OUTPUT DRC32 30
ENC NGATE CPG LT 2.0 OUTPUT DRC33N 30
ENC PGATE CPG LT 2.0 OUTPUT DRC33P 30
ENC NGATE CAA LT 3.0 OUTPUT DRC34N 30
ENC PGATE CAA LT 3.0 OUTPUT DRC34P 30
EXT [T] FIELDP CAA LT 1.0 OUTPUT DRC35 30
;*****JIM'S PERSONAL GATE LENGTH CHECK*****
;*****WIDTH GATE LT 3.0 OUTPUT DRC36 30
;*****
;***** LAYER 4 CHECKING (SELECT) *****
;*****
ENC [T] NSRCDRN CSP LT 2.0 OUTPUT DRC42 30
ENC [T] PCONT CSP LT 1.0 OUTPUT DRC43A 30
EXT NCONT CSP LT 1.0 OUTPUT DRC43B 30
WIDTH CSP LT 2.0 OUTPUT DRC44A 30
EXT CSP LT 2.0 OUTPUT DRC44B 30

```



```

;*****
;***** LAYER 5 CHECKING (POLY CONT) ***
;*****
WIDTH CCP                LT 2.0      OUTPUT      DRC5A1  30
AREA CCP                RANGE  0.0 3.9  OUTPUT      DRC5B1  30
AREA CCP                RANGE  4.1 100  OUTPUT      DRC5C1  30
;*****THE FOLLOWING TWO RULES ARE FOR DENSER***
;*****CONTACT SPACING RULES (VERY INCOMPLETE)**
ENC[T] CCP CPG          LT 1.0      OUTPUT      DRC5A2  30
EXT  CCP                LT 2.0      OUTPUT      DRC5A3  30

;*****
;***** LAYER 6 CHECKING (ACTIVE CONT) *
;*****
WIDTH CCA                LT 2.0      OUTPUT      DRC6A1  30
AREA CCA                RANGE  0.0 3.9  OUTPUT      DRC6B1  30
AREA CCA                RANGE  4.1 100  OUTPUT      DRC6C1  30
ENC[T] CCA CAA          LT 1.0      OUTPUT      DRC6A2  30
EXT  CCA                LT 2.0      OUTPUT      DRC6A3  30
EXT[T] CCA PGATE        LT 2.0      OUTPUT      DRC6P4  30
EXT[T] CCA NGATE        LT 2.0      OUTPUT      DRC6N4  30

;*****
;***** LAYER 7 CHECKING (METAL1) *****
;*****
WIDTH CME                LT 3.0      OUTPUT      DRC71   30
EXT  CME                LT 3.0      OUTPUT      DRC72   30
ENC[T] CCA CME          LT 1.0      OUTPUT      DRC73   30
ENC[T] CCP CME          LT 1.0      OUTPUT      DRC74   30

;*****
;***** LAYER 8 CHECKING (VIA) *****
;*****
WIDTH CVA                LT 2.0      OUTPUT      DRC8A1  30
AREA CVA                RANGE  0.0 3.9  OUTPUT      DRC8B1  30
AREA CVA                RANGE  4.1 100  OUTPUT      DRC8C1  30
EXT  CVA                LT 3.0      OUTPUT      DRC82   30
ENC[T] CVA CME          LT 1.0      OUTPUT      DRC83   30
EXT[T] CVA CPG          LT 2.0      OUTPUT      DRC84A  30
EXT[T] CVA CAA          LT 2.0      OUTPUT      DRC84B  30
EXT[T] CVA CCA          LT 2.0      OUTPUT      DRC85A  30
EXT[T] CVA CCP          LT 2.0      OUTPUT      DRC85B  30
EXT[T] CVA CCE          LT 2.0      OUTPUT      DRC85C  30

```

```

;*****
;***** LAYER 9 CHECKING (METAL2)*****
;*****
WIDTH CMS                LT 3.0      OUTPUT      DRC91   30
EXT   CMS                LT 4.0      OUTPUT      DRC92   30
ENC [T] CVA CMS          LT 1.0      OUTPUT      DRC93   30
;*****
;***** LAYER 10 CHECKING (OVERGLASS)**
;*****
WIDTH COG                LT 88.0     OUTPUT      DRC101  30
ENC [T] COG CMS          LT 6.0     OUTPUT      DR103A  30
ENC [T] COG CME          LT 6.0     OUTPUT      DR103B  30
EXT [N] XP CMS           LT 30.0    OUTPUT      DRC104  30
EXT [N] XP CME           LT 15.0    OUTPUT      DR105A  30
EXT [N] XP CPG           LT 15.0    OUTPUT      DR105B  30
EXT [N] XP CAA           LT 15.0    OUTPUT      DR105C  30
;*****
;***** LAYER 11 CHECKING (ELECTRODE)**
;*****
WIDTH CEL                LT 3.0      OUTPUT      DRC111  30
EXT   CEL                LT 3.0      OUTPUT      DRC112  30
ENC [T] CEL CPG          LT 3.0      OUTPUT      DRC113  30
EXT [T] CEL CMS          LT 3.0      OUTPUT      DRC114  30
EXT [T] CEL CCP          LT 3.0      OUTPUT      DRC115  30
ENC [T] CCE CEL          LT 3.0      OUTPUT      DRC116  30
;*****
;***** LAYER CCE CHECKING *****
;*****
WIDTH CCE                LT 2.0      OUTPUT      DR12A1  30
AREA  CCE                RANGE 0.0 3.9  OUTPUT      DR12B1  30
AREA  CCE                RANGE 4.1 100 OUTPUT      DR12C1  30
EXT   CCE                LT 2.0      OUTPUT      DR12A3  30
EXT [T] CCE PGATE        LT 2.0      OUTPUT      DR12P4  30
EXT [T] CCE NGATE        LT 2.0      OUTPUT      DR12N4  30

```

*END

APPENDIX 4. LVS command file

```

; LVS Command File for double Metal/double Poly CMOS Process
;
;TRANSLATE-PROCESS = //kevily/users/hansen1/des/pdf.bin
;TRANSLATE-INPUT   = //kevily/users/hansen1/des/ho
;TRANSLATE-GO      = yes
*DESCRIPTION
PRIMARY           = HO
INDISK            = \\INLVS
OUTDISK           = LVSOUP
PRINTFILE         = LVSPRT
MODE              = EXEC NO
SYSTEM            = GDS2
SCALE             = 0.01 MICRON
RESOLUTION        = 0.25 MICRON
SCHEMATIC         = LVSLOGIC
LISTERROR         = YES
KEEPDATA          = SMART
MODEL             = MOS [N] ,NCH MOS [P] ,PCH
*END
*INPUT-LAYER
CWP               = 8
CAA               = 14
CPG               = 15
CSP               = 5
CCP               = 18
CCA               = 16
CMF               = 12 TEXT 50 ATTACH = CMF
CVA               = 11
CMS               = 10
COG               = 9
CEL               = 3
CCE               = 4
CONNECT-LAY      = CWP PSRCDRN NSRCDRN CPG CMF CMS CEL
*END
*OPERATION
AND CAA CSP PDIF                                ;DEFINE ACTIVE P REGION
NOT CAA PDIF NDIF                               ;DEFINE ACTIVE N REGION
AND CPG NDIF NGATE                              ;DEFINE N GATES
AND CPG PDIF PGATE                              ;DEFINE P GATES
NOT NDIF     NGATE NSRCDRN                      ;DEFINE N SOURCE/DRAINS
NOT PDIF     PGATE PSRCDRN                      ;DEFINE P SOURCE/DRAINS
AND PSRCDRN CWP PCONT                           ;DEFINE PWELL CONTACT
AND CPG CEL PCAP                               ;POLY1-POLY2 CAP

```

```
CONNECT CME CPG      BY  CCP
CONNECT CME PSRCDRN BY  CCA
CONNECT CME NSRCDRN BY  CCA
CONNECT PSRCDRN CWP BY  PCONT
CONNECT CEL CME      BY  CCE
CONNECT CME CMS      BY  CVA
```

```
ELEMENT MOS [P] PGATE CPG PSRCDRN      ;
ELEMENT MOS [N] NGATE CPG NSRCDRN CWP  ;DEFINE P TRANSISTORS
ELEMENT CAP [PP] PCAP CPG CEL           ;DEFINE N TRANSISTORS
                                           ;POLY1-POLY2 CAP
```

```
PARAMETER CAP [PP] 5E-16
```

```
LVSCHK[X] PRINTLINE=100 WPERCENT=10 LPERCENT=10
*END
```