# VLSI architectures for implementation of neural networks

Massimo A. Sivilotti, Michael R. Emerling, and Carver A. Mead

Citation: AIP Conference Proceedings **151**, 408 (1986); doi: 10.1063/1.36247
View online: http://dx.doi.org/10.1063/1.36247
View Table of Contents:
http://scitation.aip.org/content/aip/proceeding/aipcp/151?ver=pdfcov
Published by the AIP Publishing

# VLSI Architectures for Implementation of Neural Networks

Massimo A. Sivilotti, Michael R. Emerling and Carver A. Mead[1]
California Institute of Technology, Pasadena CA 91125
April 15, 1986

## Introduction

A large scale collective system implementing a specific model for associative memory was described by Hopfield [1]. A circuit model for this operation is illustrated in Figure 1, and consists of three major components. A collection of active gain elements (called amplifiers or "neurons") with gain function $V = g(v)$ are connected by a passive interconnect matrix which provides unidirectional excitatory or inhibitory connections ("synapses") between the output of one neuron and the input to another. The strength of this interconnection is given by the conductance $G_{ij} = G_0 T_{ij}$. The requirements placed on the gain function $g(v)$ are not very severe [2], and easily met by VLSI-realizable amplifiers. The third circuit element is the capacitances that determine the time evolution of the system, and are modelled as lumped capacitances.

This formulation leads to the equations of motion shown in Figure 2, and to a Liapunov energy function which determines the dynamics of the system, and predicts the location of stable states (memories) in the case of a symmetric matrix $T$.

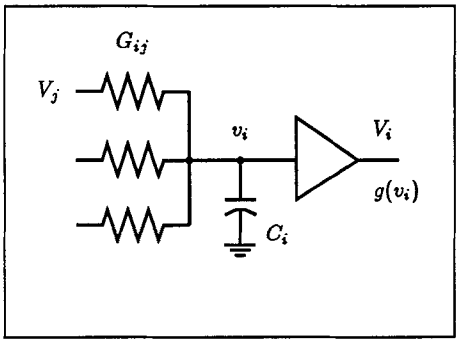


Figure 1: Circuit model for Hopfield system

$$C \cdot \frac{dv_i}{dt} = \sum_{j=1}^{N} G_{ij}(\pm V_j - v_i)$$
$$= \sum_{j=1}^{N} G_{ij}(\pm g(v_j) - v_i)$$

$$E = -\tfrac{1}{2} \sum_i \sum_j G_{ij} V_i V_j + \sum_i \left( \sum_j G_{ij} \right) \int_0^{V_i} g_i^{-1}(V) dV$$

$$\frac{\partial E}{\partial t} \leq 0 \quad \text{and} \quad \frac{\partial E}{\partial t} = 0 \Rightarrow \frac{\partial V_i}{\partial t} = 0$$

Figure 2: Differential Equations of Motion

## VLSI Restrictions

Since collective systems exhibit interesting global properties as a consequence of having large numbers of individually simple elements, implementation with very large scale integration (VLSI) circuit technology appears very suitable. There are, however, a number of technology-dependent limitations that are introduced by such a choice.

### Cost

The principal cost measure in VLSI is *area*. Even with the use of die-stitching techniques, there exists a physical limit on the maximum area a circuit can occupy. Also, the off-chip environment is quite different from the internal circuit, for electrical reasons. This fact, coupled

[1]This research was supported by the System Development Foundation

0094-243X/86/1510408-6$3.00 

with the fundamental restriction on I/O pads, makes it desirable to integrate an *entire* system on a single chip.

Analog electronics, by exploiting the intrinsic physics of native devices, generally occupy less area per function than an implementation using a digital abstraction. For example, an analog differential-input multiplier may require as few as 8 transistors to perform the relatively complex calculation $(y = k(x_{1+} - x_{1-})(x_{2+} - x_{2-}))$. Furthermore, there is none of the overhead associated with mapping what is essentially a continuous problem into a discrete-time (sampled digital) system.

## Power

A common complaint about analog computing elements is that their power consumption is high, due to a desire for maximum linearity at high operating speeds, and because discrete (off-chip) components present relatively highly capacitive loads. In a VLSI context, power dissipation must be limited to a few watts (for conventional packaging technologies). However, collective circuits implemented entirely on one die have no requirements to drive external loads, do not have to be particularly fast, and value symmetry much more highly than linearity.

It is important to note that the Hopfield circuit model exhibits non-zero power dissipation even after convergence is reached, and the computation is nominally terminated. For systems of several hundred amplifiers, it is not possible to build interconnect matrices of tens of thousands of resistors without explicitly limiting the power consumption of the amplifiers. A commonly suggested alternative, computation by current summing, is even more impractical, as the number of (power dissipating) current injectors that must be controlled scales with the number of synapses.

The approach we have taken to permit the implementation of large arrays is to limit the current consumption of the amplifiers, guaranteed by keeping most of the MOS devices in a subthreshold regime of operation [3]. For sufficiently low gate voltages (less than the so-called "threshold voltage", below which the digital abstraction of transistor operation classifies the transistor as "off"), the drain current is exponential in the gate voltage. This behavior is exactly the same (and indeed, the physics are identical) as bipolar transistors exhibit throughout their operating range, with the additional benefit that MOS transistors draw no gate current.

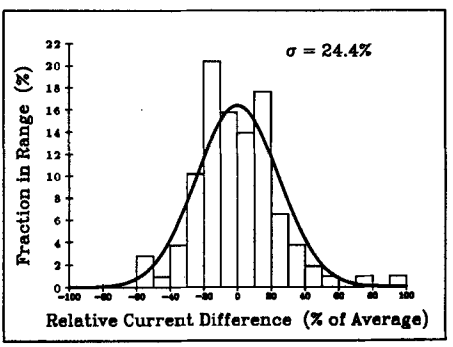


Figure 3: Full wafer current variation in 3x3 micron MOS transistors

## Parameter Variation

An additional complication is introduced in the case of very small devices, where statistical or systematic doping variations can affect their transfer characteristics by significant amounts. These variations are particularly evident in the case of fabrication lines intended for digital

chips (which are relatively insensitive to such variation). If an analog design methodology is used that requires currents to be precisely matched or subtracted, it is unlikely that sufficient accuracy can be obtained with single small transistors. The degree of variation to be expected is illustrated in Figure 3 [4], which shows the drain currents of identically biased MOS transistors from a typical MOSIS [5] digital process. When differences between *adjacent* transistor currents are taken (Figure 4), the variation in relative current differences is substantial, and indicates (in this case) a fairly random process, as opposed to some longer-range (die-scale) systematic variation. These variations can be minimized by using larger transistors (Figure 5), or by relying on statistical numbers of transistors to participate in a computation.

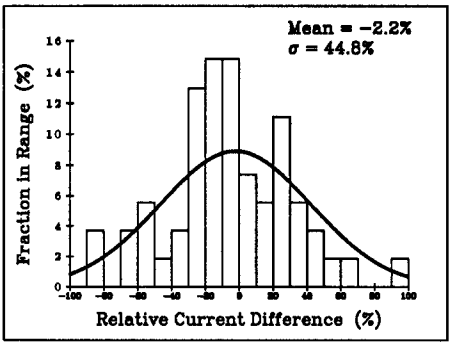


Figure 4: Variation in adjacent 3x3$\mu$ MOST

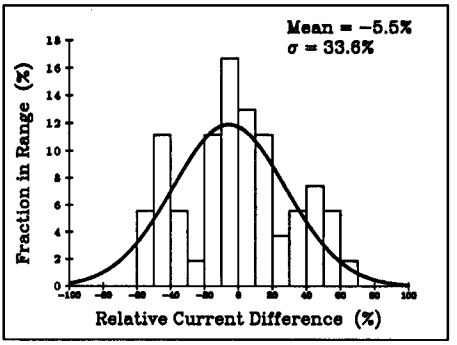


Figure 5: Variation in adjacent 24x24$\mu$ MOST

Furthermore, it is clear that a computational scheme must be designed that is robust against such variation, and that displays a high tolerance to noise. Such claims are commonly made of collective systems; they must be carefully examined, however, in light of the actual implementation.

## VLSI Implementations

The two principal implementation issues that have yet to be addressed are the requirement for negative valued resistive elements for inhibitory connections, and the desire for symmetrical large signal behaviors to maximize noise margins. Together, these considerations led to the adoption of amplifiers with differential input stages, and complementary outputs. The symmetry of the resulting dual rail signal representation makes any element with sufficient gain adequate for the amplifier. For the first implementation, we picked the simplest element possible, the nMOS inverter. There is a controllable interconnection between the inverters, which can be used to enforce to varying degrees the complementarity of the outputs. Figure 6 shows the schematic of the amplifier, with the cross-couple network implemented as two cross-connected NAND gates. Input pass transistors, marked as gated on $\phi_1$ are used to disconnect the inverters from the matrix, allowing the state of the system to be latched dynamically.

Figure 7 shows test data illustrating the transfer function of the actual amplifier. The output of the amplifier is plotted as a function of its input; all data are normalized to voltage rails of $-1$ to $+1$, and the differential nature of the inputs has been explicitly incorporated into the plot. As expected, the plot is symmetrical.

The other major component of the design is the interconnection element (Figure 8). Each $T_{ij}$ has 4 pass transistors, operating in their ohmic (resistive) regime, and is capable of 3 interconnection strengths $(-1, 0, +1)$. The $\phi_2$ line is used to selectively disconnect the matrix, and, in conjunction with the $\phi_1$ line, is used to single-step through the chip's operation.

Nothing more would be required, if it were not for the desire to dynamically change the

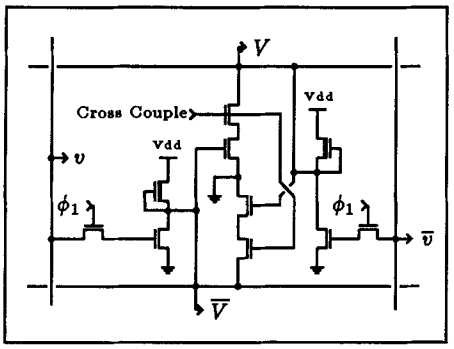


Figure 6: nMOS amplifier schematic

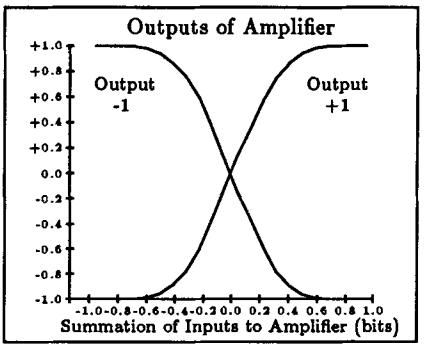


Figure 7: Amplifier transfer function

programming of the matrix. For the nMOS prototype, we opted for programmability by a generalized outer product scheme in which any two vectors may be multiplied (Figure 9). Each matrix element is stored at the corresponding interconnection element. The component of the cross product is generated at each $T_{ij}$ site (by a simple AND pass network), and is added to the previous $T_{ij}$ value, which is then stored in the tri-stable memory element.

This addition operation is truncating (addition table in Figure 10); it is *not* associative in the algebraic sense, but symmetry is preserved (if only symmetrical matrices are added). Simulation had indicated that the clipped $T_{ij}$ matrix would not greatly impact the capacity of the system as a whole; this observation has been verified experimentally.

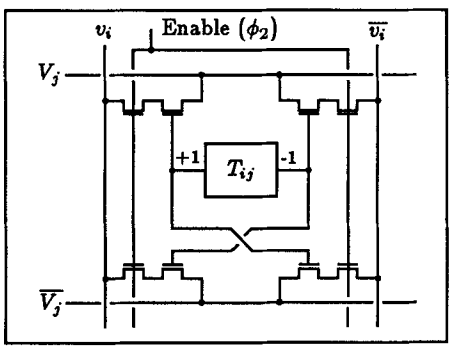


Figure 8: $T_{ij}$ Element

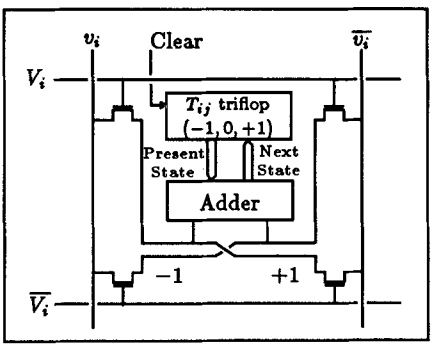


Figure 9: $T_{ij}$ Block Diagram

The entire programmable $T_{ij}$ element contained 41 transistors. At any time, only *four* of these (the dual-rail interconnect transistors in the corners of Figure 8) participate in the association process; the rest of the circuitry is dedicated to providing programmability.

## Testing the nMOS prototype

A design containing 22 active elements, and a full interconnect matrix (462 elements) was fabricated using MOSIS' 4$\mu$m feature size nMOS technology. The entire project measured 6700$\mu$m by 5700$\mu$m, and required 53 I/O pads. The completed ASSOCMEM chip is shown in Figure 11.

The chip worked immediately, with 3 memories being routinely programmable. More rarely, 4 memories were possible, for carefully chosen (i.e. nearly orthogonal) vectors. These capacities

| Outer Product \ Present State | -1 | 0 | +1 |
|---|---|---|---|
| +1 | 0 | +1 | +1 |
| 0 | -1 | 0 | +1 |
| -1 | -1 | -1 | 0 |

Next State

Figure 10: Truncating adder table

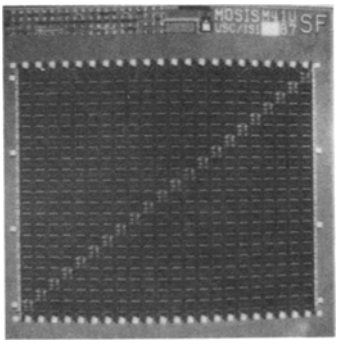

Figure 11: The ASSOCMEM

do not include the complements of the desired memories, which are themselves stable due to the symmetry of the system.

With two memories in the system, if the starting state is any closer to either of the memories, the system was found to converge directly into that stable state. Figure 12 plots the probability of falling into each of two randomly chosen memories, as a function of the Hamming distance from the initial state to the memories.

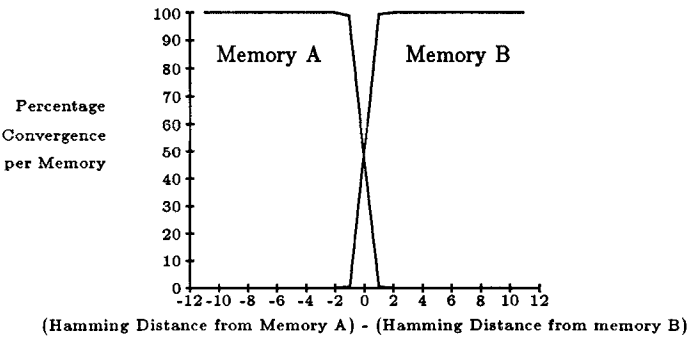


Figure 12: Plot of Convergence Probability

Figure 13 shows an incremental association in progress. By driving the $\phi_1$ and $\phi_2$ lines with a 2-phase non-overlapping clock, the inputs and outputs of the amplifiers are selectively connected to the matrix. Thus, the association can be halted, and a discrete notion of time is introduced. In the example shown, the initial state is intermediate between two memories. Within one cycle, the common bits between the two memories become active. The system is now at a metastable point, and takes 2 cycles to evolve away from it, finally settling in one of the memories.

It is worth mentioning that when run in continuous mode, we were never able to detect an association still in progress after the 50$\mu$s that our instrumentation required to switch from driving to monitoring the I/O lines.

Finally, Figure 14 deals with the issue of fault tolerance. This chip had over 40% of its $T_{ij}$ elements unprogrammable, yet could still associate to one of 2 stored states. Of the 10 chips returned by MOSIS, all had one particular element malfunction, 2 had no other elements fail, and the others ranged from 2% (10) to 12% (54) bad, excluding the chip in Figure 14. All could store at least 2 memories.

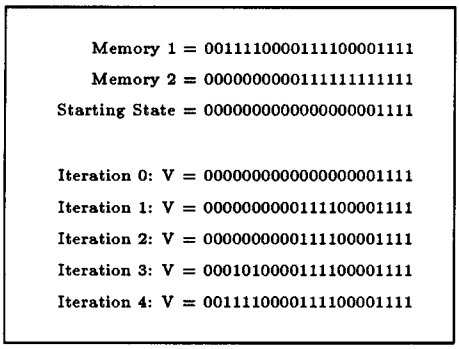


Figure 13: Incremental association

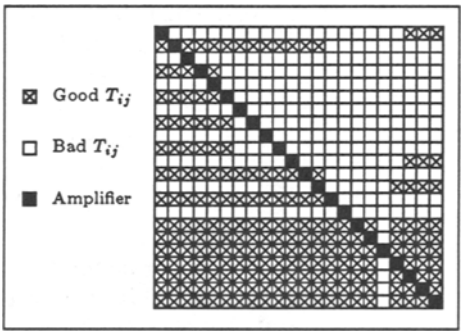


Figure 14: $T_{ij}$ yield

## Conclusions

Collective systems exhibit many appealing properties, including robustness and fault tolerance, an ability to deal with ill-posed problems and noisy data, which conventional digital architectures do not. Thus, neural computation suggests a design methodology that may permit the application of VLSI to difficult perception problems, such as vision and audition, with fault tolerance permitting wafer-scale processing surfaces.

Research is proceeding on alternative design techniques. The associative memory design has been transferred to subthreshold CMOS, and a 289 neuron chip has been fabricated. Arranged as a 17x17 pixel array, a subset of the ASCII character set has been programmed as memories. Testing is proceeding.

Finally, as feature sizes continue to shrink to the $1\mu$m range, the design of chips containing 1000 fully-interconnected neurons becomes feasible. Such chips would represent viable tools to assist in the modelling of neural networks, by providing an efficient computational implementation.

## References

[1] J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences USA*, 79:2554–2558, April 1982.

[2] J. Hopfield. Neurons with graded response have collective properties like those of two-state neurons. *Proceeding of the National Academy of Sciences USA*, 81:3088–3092, May 1984.

[3] C. Mead and M. Maher. A charge-controlled model for submicron MOS. In *Proceedings of the Colorado Microelectronics Conference*, May 1986.

[4] J. E. Tanner. *Integrated Optical Motion Detection.* PhD thesis, California Institute of Technology, 1986. In preparation.

[5] D. Cohen and G. Lewicki. MOSIS – the ARPA silicon broker. In *Proceedings from the Second Caltech Conference on VLSI*, pages 29–44, California Institute of Technology, Pasadena, CA, 1981.